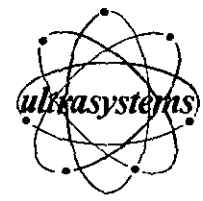
NASA CR-
147661
NAS9-14739
DRL T-476ID
LIN-003
MA-129T

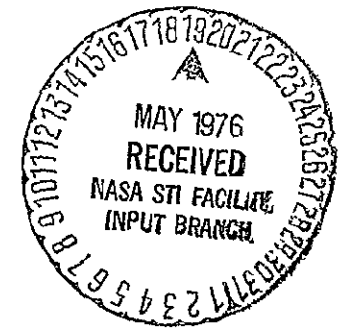


# FINAL REPORT
# ANALYSIS OF THE SURVIVABILITY OF THE SHUTTLE(ALT) FAULT-TOLERANT AVIONICS SYSTEM

APRIL 1976

APPENDICES



# ANALYSIS OF THE SURVIVABILITY OF THE SHUTTLE(ALT) FAULT-TOLERANT AVIONICS SYSTEM

## FINAL REPORT
## DATA REQUIREMENTS LIST ITEM 3
## APPENDICES

prepared for

Johnson Space Center
National Aeronautics and Space Administration
Houston, Texas 77058

Contract NAS9-14739

by

Ultrasystems, Inc.
2400 Michelson Drive
Irvine, California 92715

April 1976

76/6.43-9

FINAL REPORT

APPENDICES

TABLE OF CONTENTS



LIST OF FIGURES



LIST OF TABLES

# APPENDIX A
# ANALYTIC MODEL PROGRAM

## A.1 PROGRAM DESCRIPTION

The analytic model program calculates the Shuttle avionics survivability for the ALT configuration. The user supplies the necessary baseline parameter data to the program as well as information on parameter variation. If time is specified as the independent variable, then survivability as a function of time is generated for each parameter variation. If time is not specified as the independent variable, then survivability as a function of parameter is generated for each parameter that is specified to be varied. The only restriction on the number and type of parameters to be varied in any one run is that they must be in the same device.

The program consists of a driver routine and six subroutines. The structure of the driver program is given on the opposite page. The routine accepts the input data and checks its validity. Next, the number of time points is set up and baseline survivabilities are calculated and printed. Following this, the varying partition is identified and its survivability is set to 1. System survivability with this partition excluded may be calculated for future use. Next, the parameters are varied and the survivabilities computed and printed. A separate path is required if time is the independent variable because of differing output formats.

The actual computation is accomplished by a subroutine. There is the GPC model computed by SURVT, the MCDS model computed by MCDSC, and the flight critical bus model computed by FLTCR. The appropriate subroutine is called for the survivability calculation for each partition. The parameters associated with that partition are included in the argument list.

The printing of results is accomplished by subroutine. There are three distinct printout formats. The baseline results are printed by SVTPRT, the parameter variation results with time as the independent variable are printed by VATPRT and the parameter variation results with the parameter as the independent variable are printed by VAPPRT.

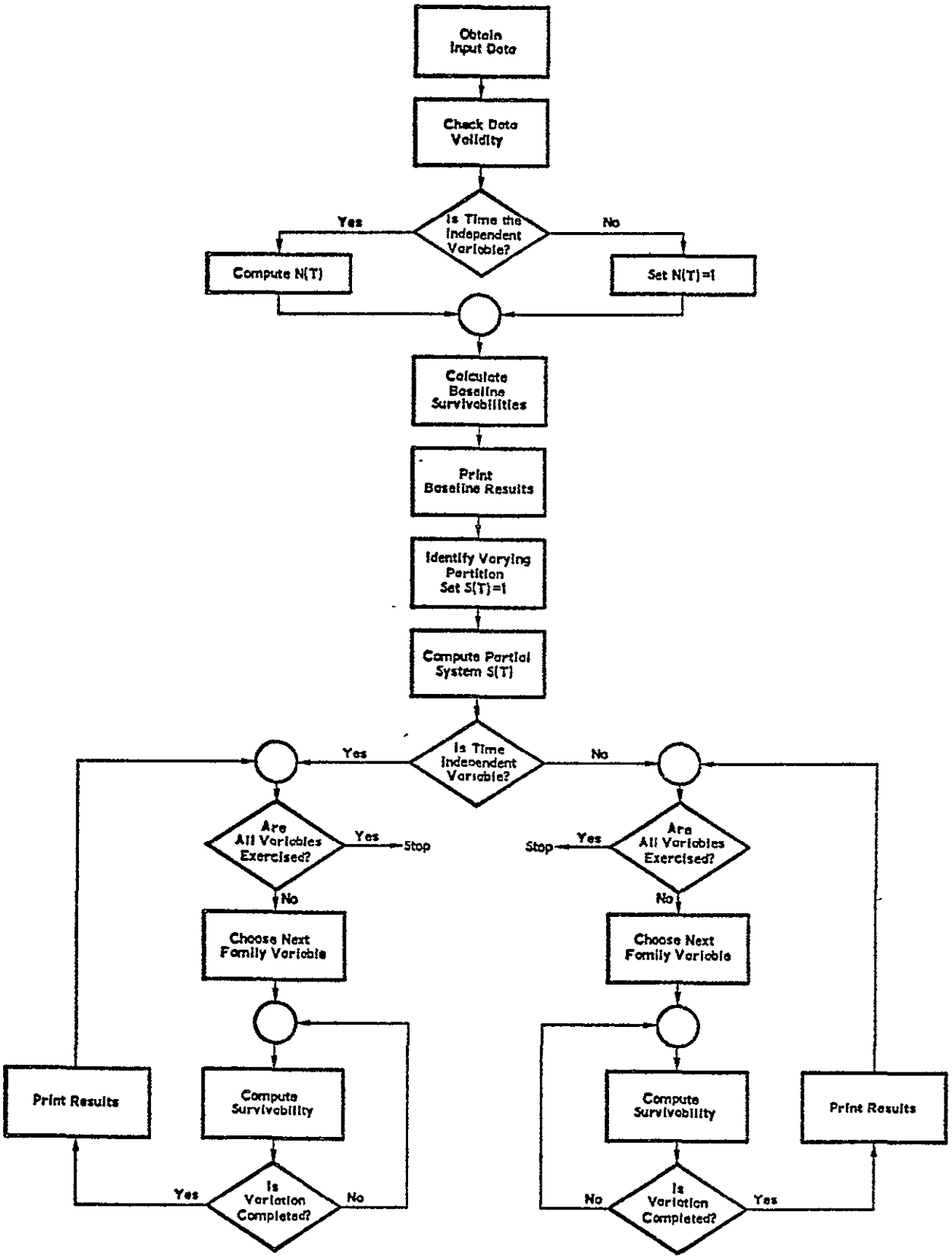



FIGURE A.1-1 OVERALL STRUCTURE OF ANALYTIC MODEL PROGRAM

## A.2. UTILIZATION PROCEDURE

### A.2.1 INTRODUCTION

Parameters are input to the program using the NAMELIST statement. This allows a relatively format free input specification and adds an identification of the parameter on the punched card. There are five namelist groups to be entered. The variables to be entered are listed in Table A.2-1, along with their group name and purpose. The variables that span several devices have an index I, while variables that apply to redundancy levels have an index K, eg. $V_2$ is detectability when 2 fault free units remain. The value of I refers to a device as shown below

| INDEX | DEVICE NAME |
|---|---|
| 1 | MCDS DEU+DU |
| 2 | MCDS KEYBOARD<br>THE ABOVE TWO INDICES REFER TO LAMDA AND TAU ONLY. FOR U,V,W, AND LEAKGE USE INDEX 2 WITH K=1 1,2, AND 3 REFERRING TO DDU+DU AND 4,5 REFERRING TO THE KEYBOARD |
| 3 | GPC |
| 4 | FLIGHT FORWARD MDM |
| 5 | ADIA |
| 6 | ACCELEROMETER |
| 7 | IMU |
| 8 | IACAN |
| 9 | MSBLS |
| 10 | RHC |
| 11 | RPTA |
| 12 | SBTC |
| 13 | FLIGHT AFT MDM |
| 14 | SERVO AMP |
| 15 | RATE GYRO |
| 16 | DDU |
| 17 | AVVI |
| 18 | A/MI |
| 19 | HST |
| 20 | ADI |
| 21 | PCM MASTER |
| 22 | OF MDM |
| 23 | OA MDM |
| 24 | FF MDM 4 |
| 25 | FA MDM 4 |

This index system assigns the input parameters and program control variables to the proper device. For example, U(4,3) is $U_4$ for the GPCs.

TABLE A.2-I INPUT VARIABLES AND THEIR PURPOSE

| VARIABLE | GROUP | PURPOSE |
|---|---|---|
| I | | Index number indicating a device |
| LAMDA(I) | PARAMS | Permanent Fault Rate |
| TAU(I) | PARAMS | Transient Fault Rate |
| LEAKAGE(K,I) | PARAMS | Transient Leakage |
| U(K,I) | PARAMS | Detectability |
| V(K,I) | PARAMS | Diagnosability |
| W(K,I) | PARAMS | Recoverability |
| DELT | MSNT | Time Increment |
| TMAX | MSNT | Maximum value of time |
| VARPAR | MSNT | Device number to vary parameters |
| ITER | MSNT | ITER 2 allows another run to follow |
| TIME | FLAGS | Indicates time is the independent variable |
| UFLG(K) | FLAGS | Indicates U(K,VARPAR) is to be varied |
| VFLG(K) | FLAGS | Indicates V(K,VARPAR) is to be varied |
| WFLG(K) | FLAGS | Indicates W(K,VARPAR) is to be varied |
| LKGFLG(K) | FLAGS | Indicates LEAKGE(K,VARPAR) is to be varied |
| LAMFLG | FLAGS | Indicates LAMDA(VARPAR) is to be varied |
| TAUFLG | FLAGS | Indicates TAU(VARPAR) is to be varied |
| DELU(K) | DELS | Increment in U(K,VARPAR) |
| DELV(K) | DELS | Increment in V(K,VARPAR) |
| DELW(K) | DELS | Increment in W(K,VARPAR) |
| DELLKG(K) | DELS | Increment in LEAKGE(K,VARPAR) |
| DELLAM | DELS | Increment in LAMDA(VARPAR) |
| DELTAU | DELS | Increment in TAU(VARPAR) |
| UMAX(K) | MAXS | Maximum value of U(K,VARPAR) |
| VMAX(K) | MAXS | Maximum value of V(K,VARPAR) |
| WMAX(K) | MAXS | Maximum value of W(K,VARPAR) |
| LKGMAX(K) | MAXS | Maximum value of LEAKGE(K,VARPAR) |
| LAMMAX | MAXS | Maximum value of LAMDA(VARPAR) |
| TAUMAX | MAXS | Maximum value of TAU(VARPAR) |
| TVAL | MAXS | Time when not the independent variable |

### A.2.2 INPUT DECK SETUP

Within the program, some parameters are assigned default values. That is, if a parameter is not input, it is assigned a preset value. All Us are assigned .999. All Vs and Ws are assigned 1 except V(2,I) is assigned a value of .95. LAMDAs and TAUs are 100 (per $10^6$ hours). All FLAGS are .FALSE. Leakages are zero.

A sample input deck set-up is shown in Figure A.2-1. The namelist groups are to come in the following order: PARAMS, MSNT, FLAGS, DELS, MAXS. The initial card for each group must begin with a dollar sign ($) in column 2 immediately followed by the group name with no imbedded blanks. Succeeding data items are read until a $ is encountered. Each data item is separated by commas and may be in any of two forms:

1. A variable equals a constant
2. An array with or without subscript followed by a list of constants, separated by commas.

In (2), the subscript indicates the beginning location for the assignment. No subscript implies a 1. A variable may be omitted or may have more than one assignment. A value of ITER greater than 2 allows another input deck to be run after the current one.

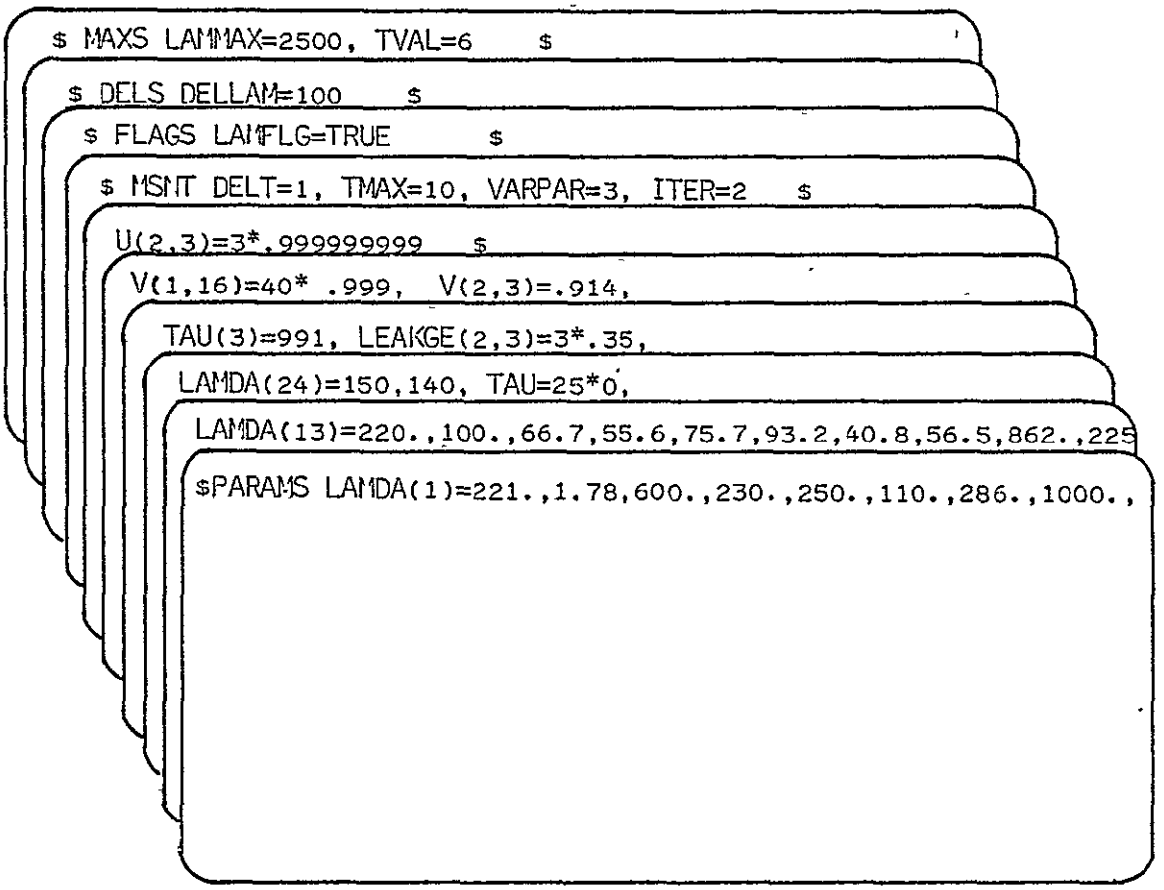



FIGURE A.2-1 A SAMPLE INPUT DECK SET-UP

```
      PROGRAM A(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C
C          EACH UNIT OF THE ALT OP SYSTEM IS REPRESENTED BY AN INDEX.
C        THIS IS ALIST OF THESE REPRESENTATIONS.
C
C
C  INDEX     UNIT REPRESENTED
C
C     1      MCDS DEU+DU
C     2      MCDS KEYBOARD
C                  ABOVE TWO REFER TO LAMDA AND TAU  ONLY.
C                  FOR U,V,W,AND LEAKGE USE INDEX 2 WITH POSITIONS
C                  1,2, AND 3 REFERRING TO DDU+DU AND 4,5 REFERRING
C                  TO THE KEYBOARD
C     3      GPC
C     4      FLIGHT FORWARD MDM
C     5      ADTA
C     6      ACCELEROMETER
C     7      IMU
C     8      TACAN
C     9      MSBLS
C     10     RHC
C     11     RPTA
C     12     SBTC
C     13     FLIGHT AFT MDM
C     14     SERVO AMP
C     15     RATE GYRO
C     16     DDU
C     17     AVVI
C     18     A/MI
C     19     HSI
C     20     ADI
C     21     PCM MASTER
C     22     OF MDM
C     23     OA MDM
C     24     FF MDM 4
C     25     FA MDM 4
C
C
C              VARIABLE DICTIONARY
C
C       ABTFLG        ABORT FLAG, A LOGICAL VARIABLE
C       ALPHA(J,K)    COEFFICIENTS OF SURVIVABILITY EQUATION
C       B(I)          LOGICAL VARIABLE USED IN INPUT VALIDITY CHECK
C       COVRGE(K)     COVERAGE
C       DELT          TIME INCREMENT
C       DELTA(K)      PERMANENT FAULT RATE PLUS LEAKY TRANSIENT RATE
C       LAMDA         PERMANENT FAULT RATE IN FAULTS PER HOUR
C       LAMFLG        FLAG  INDICATING(IF TRUE) DESIRE TO VARY LAMDA
C       LAMMAX        MAXIMUM VALUE OF LAMDA
C       LEAKGE(K)     PERCENTAGE OF TRANSIENTS MISTAKEN FOR PERMANENT
C       LKGFLG(K)     FLAG  INDICATING(IF TRUE) DESIRE TO VARY LKGFLG(K)
C       LKGMAX        MAXIMUM VALUE OF LEAKAGE
C       M             LOOP INDEX IN SURVIVABILITY CALCULATION
```

ANALYTIC MODELING PROGRAM PRINTOUT

```
      C     N           NUMBER OF COMPUTERS IN SYSTEM
      C     ORGVAL      AN ARRAY USED TO STORE THE INITIAL VALUES OF U,V,H,
      C                    LEAKGE,LAMDA AND TAU
      C     SIGMA(K)    SUM OF PERMANENT FAULT RATE AND LEAKAGE/TRANSIENT FAULT
60    C                    RATE PRODUCT.
      C     SIGMAT      SUM OF THE PERMANENT AND TRANSIENT FAULT RATES.  EXPRESSED
      C                    IN FAULTS PER HOUR
      C     SUM         TEMPORARY VARIABLE FOR ALPHA COEFFICIENT CALCULATION
      C     SURVIV(M)   SURVIVABILITY OF UNIT OF INTEREST AT MTH TIME INTERVAL
65    C     T(M)        TIME
      C     TAU         TRANSIENT FAULT RATE IN FAULTS PER HOUR
      C     TAUFLG      FLAG  INDICATING(IF TRUE) DESIRE TO VARY TAU
      C     TAUMAX      MAXIMUM VALUE OF TAU
      C     TEMP        TEMPORARY VARIABLE IN SURVIVABILITY CALCULATION
70    C     TEMP1       TEMPORARY VARIABLE IN TMAX/DELT VALIDITY CHECK
      C     TIME        FLAG INDICATING(IF TRUE) TIME AS THE INDEPENDENT VARIABLE
      C     TMAX        MAXIMUM TIME FOR WHICH SURVIVABILITY IS CALCULATED
      C     TVAL        THE VALUE OF TIME FOR WHICH COMPUTATIONS WILL BE MADE
      C                    USING ANOTHER INDEPENDENT VARIABLE
75    C     U(K)        DETECTABILITY
      C     UFLG(K)     FLAG  INDICATING(IF TRUE) DESIRE TO VARY U(K)
      C     UMAX        MAXIMUM VALUE OF DETECTABILITY
      C     V(K)        DIAGNOSABILITY
      C     VFLG(K)     FLAG  INDICATING(IF TRUE) DESIRE TO VARY V(K)
80    C     VMAX        MAXIMUM VALUE OF DIAGNOSABILITY
      C     W(K)        RECOVERABILITY
      C     WFLG(K)     FLAG  INDICATING(IF TRUE) DESIRE TO VARY W(K)
      C     WMAX        MAXIMUM VALUE OF RECOVERABILITY
      C
85    C
      LOGICAL B(25),ABTFLG,UFLG(5),VFLG(5),WFLG(5),
     *        LKGFLG(5),LAMFLG,TAUFLG,TIME,ODUFLG(5)
      REAL      DELU(5),DELV(5),DELW(5),DELLKG(5),DELLAM,DELTAU,
     *          UMAX(5),VMAX(5),WMAX(5),LKGMAX(5),LAMMAX,TAUMAX,TVAL,
90    *      PARAM(20),RESULT(20),FCBUS(20),
     *       U(5,25),V(5,25),W(5,25),LEAKGE(5,25),T(20),
     *         ORGVAL(25),LAMDA(25),TAU(25),SURVIV(20,28)
      INTEGER HEADER(6),NR(25),VARPAR,UNITN(20)
      NAMELIST/PARAMS/LAMDA,TAU,LEAKGE,U,V,W,N /MSNT/DELT,TMAX,VARPAR
95   *    ,ITER
      NAMELIST/FLAGS/UFLG,VFLG,WFLG,LKGFLG,LAMFLG,TAUFLG,TIME
      NAMELIST/DELS/DELU,DELV,DELW,DELLKG,DELLAM,DELTAU
      NAMELIST/MAXS/UMAX,VMAX,WMAX,LKGMAX,LAMMAX,TAUMAX,TVAL
      DATA UNITN/6HMCOSOU,6HMCOSKB,6HGPC   ,6HFF MDM,6HADTA  ,6HACCEL ,
100  *           6HINU   ,6HTACAN ,6HMSBLS ,6HRHC   ,6HRPIA  ,6HSBTC  ,
     *           6HFA MDM,6HASA   ,6HRGYRO ,6HDOU   ,6HAVVI  ,
     *           6HA/MI  ,6HHSI   ,6HADI   ,6HPCMMU ,6HOF MDM,6HOA MDM,
     *           6HAFT FC,6HS CRIT,6HM CRIT,6HFT DIS,6HFMD FC/
     *    , NR   /3,2,3*4,7*3,2*4,3,10*2/
105  *  ,HEADER/6H ONE  ,6H TWO  ,6HTHREE ,6H FOUR ,6H FIVE ,1H  /
     *   ,LAMDA,TAU/50*100/,LEAKGE/125*0,/,U,V,W /125*.999,250*1,/
     *   ,TVAL, DELT,TMAX,UMAX,VMAX,WMAX,DELU,DELW,DELV,DELLAM,DELTAU,
     *    LAMMAX,TAUMAX/37*1.0/
      DO 5 I=1,25
110  5 V(2,I) = .95
```

```
      VARPAR=0
      DELT=1
    1 ITER = 1
      DATA UFLG,VFLG,HFLG,LKGFLG,LAMFLG,TAUFLG,TIME/23*.FALSE./
      TIME=.FALSE.
      UNITN(2)=6HNCQSKB
      DO 8 I =1,25
      LAMDA(I) = LAMDA(I)/ 1.E-6
      TAU(I)   =  TAU(I)  / 1.E-6
    8 CONTINUE
      DELLAM   =  DELLAM  / 1.E-6
      DELTAU   =  DELTAU  / 1.E-6
      LAMMAX   =  LAMMAX  / 1.E-6
      TAUMAX   =  TAUMAX  / 1.E-6
      READ(05,PARAMS)
      READ(05,MSNT)
      READ(05,FLAGS)
      READ(05,DELS)
      READ(05,MAXS)
      DO 10 I=1,25
      LAMDA(I) = LAMDA(I)* 1.E-6
      TAU(I)   =  TAU(I)  * 1.E-6
   10 CONTINUE
      DELLAM   =  DELLAM  * 1.E-6
      DELTAU   =  DELTAU  * 1.E-6
      LAMMAX   =  LAMMAX  * 1.E-6
      TAUMAX   =  TAUMAX  * 1.E-6
C
C
C         INPUT VARIABLE VALIDITY CHECK
C
C     SET B(I) FALSE   AND CHECK INPUT VARIABLES
C
      ABTFLG=.FALSE.
      DO 115 J=1,25
      DO 100 I=1,25,1
         B(I)=.FALSE.
  100 CONTINUE
      DO 110 I=1,5,1
      IF (U(I,J).GT.1..OR.U(I,J).LT.0.) B(I)=.TRUE.
         IF(B(I))          WRITE(06,101) I,U(I,J)
  101 FORMAT(1H ,10X, 2HU(,I1, 4H) = ,F10.8)
      IF (V(I,J).GT.1..OR.V(I,J).LT.0.) B(I+5)=.TRUE.
         IF(B(I+5))          WRITE(06,102) I,V(I,J)
  102 FORMAT(1H ,10X, 2HV(,I1, 4H) = ,F10.8)
      IF (H(I,J).GT.1..OR.H(I,J).LT.0.) B(I+10)=.TRUE.
         IF(B(I+10))          WRITE(06,103) I,H(I,J)
  103 FORMAT(1H ,10X, 2HW(,I1, 4H) = ,F10.8)
      IF (LEAKGE(I,J).GT.1..OR.LEAKGE(I,J).LT.0.) B(I+15)=.TRUE.
         IF(B(I+15))          WRITE(06,104) I,LEAKGE(I,J)
  104 FORMAT(1H ,10X, 7HLEAKGE(,I1, 4H) = ,F10.8)
  110 CONTINUE
      TEMP1 = TMAX/DELT
      IF((TMAX/DELT).GT.20.) B(21)=.TRUE.
      IF(B(21))  WRITE(06,111) TEMP1
```

```
  111 FORMAT(1H ,12HTMAX/DELT = ,F10.0)
C
C     IF ANY OF THE LOGICAL VARIABLES DETERMINED ABOVE HAVE BEEN SET,
C        THEN ABORT THE RUN.
C
      DO 120 I=1,25,1
         IF(A(I))            ABTFLG = .TRUE.
  120 CONTINUE
  115 CONTINUE
      IF(ABTFLG)            GO TO 999
C
C
C         FIND NT
C
      NT=1
      IF (.NOT.TIME) GO TO 400
      NT = INT(TMAX/DELT)
  400 CONTINUE
      IF (NT.EQ.1) DELT=TVAL
      DO 116 I=1,NT
      T(I)=I*DELT
  116 CONTINUE
C
C        CALCULATE BASELINE SURVIVABILITIES
C
       DO 114 I=1,2
      U      (3*I,1)=U      (I,2)
      V      (3*I,1)=V      (I,2)
      W      (3*I,1)=W      (I,2)
      LEAKGE(3*I,1)=LEAKGE(I,2)
  114 CONTINUE
      CALL MCOSC(LAMDA,TAU,U,V,W,LEAKGE,T,SURVIV(1,2),NT)
      DO 117 I=3,23
      CALL SURVT(LAMDA(I),TAU(I),U(1,I),V(1,I),W(1,I),LEAKGE(1,I)
     *          ,T,SURVIV(1,I),NR(I),NT)
  117 CONTINUE
      CALL FLTGR(LAMDA(4),TAU(4),U(1,4),V(1,4),W(1,4),LEAKGE(1,4),
     *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
      DO 118 I=1,28
      SURVIV(I,25)=1
      SURVIV(I,26)=1
  118 CONTINUE
C
C     COMPUTE SYSTEM SURVIVABILITIES
C
      DO 119 J=1,NT
      DO 121  I=2,3
      SURVIV(J,25)=SURVIV(J,25)*SURVIV(J,I)
  121 CONTINUE
      SURVIV(J,25)=SURVIV(J,25)*SURVIV(J,24)
      DO 1210 I=27,28
      SURVIV(J,25)=SURVIV(J,25)*SURVIV(J,I)
 1210 CONTINUE
      DO 122 I=21,23
      SURVIV(J,26)=SURVIV(J,26)*SURVIV(J,I)
```

```
      122 CONTINUE
          SURVIV(J,26)=SURVIV(J,26)*SURVIV(J,25)
      124 CONTINUE
C
C         PRINT BASELINE RESULTS
C
          CALL SVIPRT(NI,LAMDA,TAU,U,V,W,T,SURVIV,NR,LEAKGE,UNITH)
C
C     PRESERVE ORIGINAL VALUES
C
          IF (VARPAR.LE.0) GO TO 998
          DO 130 I=1,5,1
              ORGVAL(I) = U(I,VARPAR)
              ORGVAL(I+5) = V(I,VARPAR)
              ORGVAL(I+10) = W(I,VARPAR)
              ORGVAL(I+15) = LEAKGE(I,VARPAR)
      130 CONTINUE
          ORGVAL(21) = LAMDA (VARPAR)
          ORGVAL(22) = TAU (VARPAR)
          DO 131 I=1,5
          U(I,2) = U(I,1)
          V(I,2) = V(I,1)
          W(I,2) = W(I,1)
          LEAKGE(I,2) = LEAKGE(I,1)
      131 CONTINUE
          NJ = NR(VARPAR)
          J = VARPAR
          IF (VARPAR.LE.2) NJ=5
          JS = VARPAR
          IF (VARPAR.EQ.1) JS=2
C
C         REMOVE VARYING PARTITION FROM S(T) FOR SYSTEM
C
          JF = JS
          IF (JS .GE. 4 .AND. JS .LE. 9) JF = 28
          IF (JS .GE. 10 .AND. JS .LE. 12) JF = 24
          IF (JS .GE. 13 .AND. JS .LE. 20) JF = 27
           DO 123 I=1,NT
          SURVIV(I,26) = SURVIV(I,26) / SURVIV(I,JF)
          IF (VARPAR .LE. 20) SURVIV(I,25) = SURVIV(I,25) / SURVIV(I,JF)
      123 CONTINUE
          IF(TIME)
C             THEN
         *GO TO 1001
C             ELSE
          GO TO 2000
C        SELECTION OF VARIABLE FOR WHICH FAMILY OF DATA IS TO
C            BE COMPUTED
C CHOOSE1
 1001 DO 201 I=1,NJ
          IF (.NOT.UFLG(I)) GO TO 201
      211 CONTINUE
          IF (VARPAR.LE.2) GO TO 501
          CALL SURVTI(LAMDA(J),TAU(J),U(1,J),V(1,J),W(1,J),LEAKGE(1,J)
         *           ,T,SURVIV(1,J),NR(J),NT)
```

```
      IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 503
      GO TO 502
  503 CONTINUE
      CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),W(1,4),LEAKGE(1,4),
     *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
      GO TO 502
  501 CONTINUE
      CALL MCOSC(LAMDA,TAU,U,V,W,LEAKGE,T,SURVIV(1,2),NT)
  502 CONTINUE
      CALL VATPRT(NR(J),LAMDA(J),TAU(J),U(1,J),V(1,J),W(1,J),T,
     *  SURVIV,SURVIV(1,25),SURVIV(1,26),NT,UNITN(J),LEAKGE(1,J),JS,JF)
      U(I,VARPAR) = U(I,VARPAR) + DELU(I)
      IF (U(I,VARPAR).LE.UMAX(I))GO TO 211
      U(I,VARPAR) = ORGVAL(I)
      UFLG(I) = .FALSE.
  201 CONTINUE
      DO 202 I=1,NJ
      IF (.NOT.VFLG(I)) GOTO 202
  212 CONTINUE
      IF (VARPAR.LE.2) GO TO 511
      CALL SURVT(LAMDA(J),TAU(J),U(1,J),V(1,J),W(1,J),LEAKGE(1,J)
     *          ,T,SURVIV(1,J),NR(J),NT)
      IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 513
      GO TO 512
  513 CONTINUE
      CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),W(1,4),LEAKGE(1,4),
     *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
      GO TO 512
  511 CONTINUE
      CALL MCOSC(LAMDA,TAU,U,V,W,LEAKGE,T,SURVIV(1,2),NT)
  512 CONTINUE
      CALL VATPRT(NR(J),LAMDA(J),TAU(J),U(1,J),V(1,J),W(1,J),T,
     *  SURVIV,SURVIV(1,25),SURVIV(1,26),NT,UNITN(J),LEAKGE(1,J),JS,JF)
      V(I,VARPAR) = V(I,VARPAR) + DELV(I)
      IF (V(I,VARPAR).LE.VMAX(I))   GO TO 212
      V(I,VARPAR) = ORGVAL(I+5)
      VFLG(I) = .FALSE.
  202 CONTINUE
      DO 203 I=1,NJ
      IF (.NOT.WFLG(I))  GOTO 203
  213 CONTINUE
      IF (VARPAR.LE.2) GO TO 521
      CALL SURVT(LAMDA(J),TAU(J),U(1,J),V(1,J),W(1,J),LEAKGE(1,J)
     *          ,T,SURVIV(1,J),NR(J),NT)
      IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 523
      GO TO 522
  523 CONTINUE
      CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),W(1,4),LEAKGE(1,4),
     *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
      GO TO 522
  521 CONTINUE
      CALL MCOSC(LAMDA,TAU,U,V,W,LEAKGE,T,SURVIV(1,2),NT)
  522 CONTINUE
      CALL VATPRT(NR(J),LAMDA(J),TAU(J),U(1,J),V(1,J),W(1,J),T,
     *  SURVIV,SURVIV(1,25),SURVIV(1,26),NT,UNITN(J),LEAKGE(1,J),JS,JF)
```

```
      W(I,VARPAR) = W(I,VARPAR) + DELW(I)
      IF (W(I,VARPAR).LE.WMAX(I))  GO TO 213
      W(I,VARPAR) = ORGVAL(I+10)
      WFLG(I) = .FALSE.
  213 CONTINUE
      DO 214 I=1,NJ
      IF (.NOT.LKGFLG(I)) GO TO 204
  214 CONTINUE
      IF (VARPAR.LE.2) GO TO 531
      CALL SURVT(LAMDA(J),TAU(J),U(1,J),V(1,J),W(1,J),LEAKGE(1,J)
     *          ,T,SURVIV(1,J),NR(J),NT)
      IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 533
      GO TO 532
  533 CONTINUE
      CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),W(1,4),LEAKGE(1,4),
     *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
      GO TO 532
  531 CONTINUE
      CALL MCOSC(LAMDA,TAU,U,V,W,LEAKGE,T,SURVIV(1,2),NT)
  532 CONTINUE
      CALL VATPRT(NR(J),LAMDA(J),TAU(J),U(1,J),V(1,J),W(1,J),T,
     *  SURVIV,SURVIV(1,25),SURVIV(1,26),NT,UNITM(J),LEAKGE(1,J),JS,JF)
      LEAKGE(I,VARPAR) = LEAKGE(I,VARPAR) + DELLKG(I)
      IF (LEAKGE (I,VARPAR).LE .LKGMAX(I)) GO TO 214
      LEAKGE(I,VARPAR) = ORGVAL(I+15)
      LKGFLG(I) = .FALSE.
  204 CONTINUE
      IF (.NOT.LAMFLG) GO TO 205
  215 CONTINUE
      IF (VARPAR.LE.2) GO TO 541
      CALL SURVT(LAMDA(J),TAU(J),U(1,J),V(1,J),W(1,J),LEAKGE(1,J)
     *          ,T,SURVIV(1,J),NR(J),NT)
      IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 543
      GO TO 542
  543 CONTINUE
      CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),W(1,4),LEAKGE(1,4),
     *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
      GO TO 542
  541  CONTINUE
      CALL MCOSC(LAMDA,TAU,U,V,W,LEAKGE,T,SURVIV(1,2),NT)
  542  CONTINUE
      CALL VATPRT(NR(J),LAMDA(J),TAU(J),U(1,J),V(1,J),W(1,J),T,
     *  SURVIV,SURVIV(1,25),SURVIV(1,26),NT,UNITM(J),LEAKGE(1,J),JS,JF)
      LAMDA(VARPAR) = LAMDA(VARPAR) + DELLAM
      IF (LAMDA(VARPAR).LE.LAMMAX) GO TO 215
      LAMDA(VARPAR) =  ORGVAL(21)
      LAMFLG = .FALSE.
  205 IF (.NOT.TAUFLG) GO TO 998
  216 CONTINUE
      IF (VARPAR.LE.2) GO TO 551
      CALL SURVT(LAMDA(J),TAU(J),U(1,J),V(1,J),W(1,J),LEAKGE(1,J)
     *          ,T,SURVIV(1,J),NR(J),NT)
      IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 553
      GO TO  552
  553 CONTINUE
```

```
                CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),H(1,4),LEAKGE(1,4),
               *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
                GO TO 552
              551  CONTINUE
390             CALL HCOSC(LAMDA,TAU,U,V,H,LEAKGE,T,SURVIV(1,2),NT)
              552  CONTINUE
                CALL VATPRT(NR(J),LAMDA(J),TAU(J),U(1,J),V(1,J),H(1,J),T,
               *  SURVIV,SURVIV(1,25),SURVIV(1,26),NT,UNITN(J),LEAKGE(1,J),JS,JF)
                TAU(VARPAR) = TAU(VARPAR) + DELTAU
395             IF (TAU(VARPAR).LE.TAUMAX) GO TO 216
                TAU(VARPAR) = ORGVAL(22)
                TAUFLG = .FALSE.
                 GO TO 998
            C CHOOSE2
400          2000 DO 301 I=1,NJ
                IF(.NOT.UFLG(I)) GO TO 301
                M = 0
              311 CONTINUE
                M = M + 1
405             IF (VARPAR.LE.2) GO TO 505
                CALL SURVT(LAMDA(J),TAU(J),U(1,J),V(1,J),H(1,J),LEAKGE(1,J)
               *          ,T,SURVIV(1,J),NR(J),NT)
                IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 507
                GO TO 506
410           507 CONTINUE
                CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),H(1,4),LEAKGE(1,4),
               *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
                GO TO 506
              505 CONTINUE
415             CALL HCOSC(LAMDA,TAU,U,V,H,LEAKGE,T,SURVIV(1,2),NT)
              506 CONTINUE
                PARAM(M) = U(I,J)
                RESULT(M) = SURVIV(1,JS)
                FCOUS(M) = SURVIV(1,JF)
420             U(I,VARPAR) = U(I,VARPAR) + DELU(I)
                IF (U(I,VARPAR).LE.UMAX(I))GO TO 311
                U(I,VARPAR) = ORGVAL(I)
                CALL VAPPRT(PARAM,RESULT,T(1),M,UNITN(J),SURVIV(1,25),SURVIV(1,26)
               *  ,HEADER(I),6HDETECT,6HABILIT,2HY ,FCOUS,JF)
425             UFLG(I) = .FALSE.
              301 CONTINUE
                DO 302 I=1,NJ
                IF (.NOT.VFLG(I)) GO TO 302
                M = 0
430           312 CONTINUE
                M = M + 1
                IF (VARPAR.LE.2) GO TO 515
                CALL SURVT(LAMDA(J),TAU(J),U(1,J),V(1,J),H(1,J),LEAKGE(1,J)
               *          ,T,SURVIV(1,J),NR(J),NT)
435             IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 517
                GO TO 516
              517 CONTINUE
                CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),H(1,4),LEAKGE(1,4),
               *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
440             GO TO 516
```

```
      515 CONTINUE
          CALL MCOSC(LAMDA,TAU,U,V,H,LEAKGF,T,SURVIV(1,2),NT)
      516 CONTINUE
          PARAM(M) = V(I,J)
445       RESULT(M) = SURVIV(1,JS)
          FCBUS(M) = SURVIV(1,JF)
          V(I,VARPAR) = V(I,VARPAR) + DELV(I)
          IF (V(I,VARPAR).LE.VMAX(I))  GO TO 312
          V(I,VARPAR) = ORGVAL(I+5)
450       CALL VAPPRT(PARAM,RESULT,T(1),M,UNITN(J),SURVIV(1,25),SURVIV(1,26)
         *  ,HEADER(I),6HDIAGNO,6HSABILI,2HTY,FCBUS,JF)
          VFLG(I) = .FALSE.
      302 CONTINUE
          DO 303 I=1,NJ
455       IF (.NOT.HFLG(I)) GO TO 303
          M = 0
      313 CONTINUE
          M = M + 1
          IF (VARPAR.LE.2) GO TO 525
460       CALL SURVT(LAMDA(J),TAU(J),U(1,J),V(1,J),H(1,J),LEAKGE(1,J)
         *          ,T,SURVIV(1,J),NR(J),NT)
          IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 527
          GO TO 526
      527 CONTINUE
465       CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),H(1,4),LEAKGE(1,4),
         *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
          GO TO 526
      525 CONTINUE
          CALL MCOSC(LAMDA,TAU,U,V,H,LEAKGE,T,SURVIV(1,2),NT)
470   526 CONTINUE
          PARAM(M) = H(I,J)
          RESULT(M) = SURVIV(1,JS)
          FCBUS(M) = SURVIV(1,JF)
          H(I,VARPAR) = H(I,VARPAR) + DELH(I)
475       IF (H(I,VARPAR).LE.HMAX(I))  GO TO 313
          H(I,VARPAR) = ORGVAL(I+10)
          CALL VAPPRT(PARAM,RESULT,T(1),M,UNITN(J),SURVIV(1,25),SURVIV(1,26)
         *  ,HEADER(I),6HRECOVE,6HRABILI,2HTY,FCBUS,JF)
          HFLG(I) = .FALSE.
480   303 CONTINUE
          DO 304 I=1,NJ
          IF (.NOT.LKGFLG(I)) GO TO 304
          M = 0
      314 CONTINUE
485       M = M + 1
          IF (VARPAR.LE.2) GO TO 535
          CALL SURVT(LAMDA(J),TAU(J),U(1,J),V(1,J),H(1,J),LEAKGE(1,J)
         *          ,T,SURVIV(1,J),NR(J),NT)
          IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 537
490       GO TO 536
      537 CONTINUE
          CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),H(1,4),LEAKGE(1,4),
         *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
          GO TO 536
495   535 CONTINUE
```

```
      CALL MCOSC(LAMDA,TAU,U,V,H,LEAKGE,T,SURVIV(1,2),NT)
  536 CONTINUE
      PARAM(M) = LEAKGE(I,J)
      RESULT(M) = SURVIV(1,JS)
  500 FCBUS(M) = SURVIV(1,JF)
      LEAKGE(I,VARPAR) = LEAKGE(I,VARPAR) + DELLKG(I)
      IF (LEAKGE(I,VARPAR).LE.LKGMAX(I)) GO TO 314
      LEAKGE(I,VARPAR) = ORGVAL(I+15)
      LKGFLG(I) = .FALSE.
  505 CALL VAPPRT(PARAM,RESULT,T(1),M,UNITN(J),SURVIV(1,25),SURVIV(1,26)
     *  ,HEADER(I),6H   LEA,6HKAGE  ,2H  ,FCBUS,JF)
  304 CONTINUE
      IF (.NOT.LAMFLG) GO TO 305
      M = 0
  510 315 CONTINUE
      M = M + 1
      IF (VARPAR.LE.2) GO TO 545
      CALL SURVT(LAMDA(J),TAU(J),U(1,J),V(1,J),H(1,J),LEAKGE(1,J)
     *          ,T,SURVIV(1,J),NR(J),NT)
  515 IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 547
      GO TO 546
  547 CONTINUE
      CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),H(1,4),LEAKGE(1,4),
     *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
  520 GO TO 546
  545 CONTINUE
      CALL MCOSC(LAMDA,TAU,U,V,H,LEAKGE,T,SURVIV(1,2),NT)
  546 CONTINUE
      PARAM(M) = LAMDA(J)
  525 RESULT(M) = SURVIV(1,JS)
      FCBUS(M) = SURVIV(1,JF)
      LAMDA(VARPAR) = LAMDA(VARPAR) + DELLAM
      IF (LAMDA(VARPAR).LE.LAMMAX) GO TO 315
      LAMDA(VARPAR) = ORGVAL(21)
  530 CALL VAPPRT(PARAM,RESULT,T(1),M,UNITN(J),SURVIV(1,25),SURVIV(1,26)
     *  ,HEADER(6),6H    LA,6HMBDA  ,2H  ,FCBUS,JF)
      LAMFLG = .FALSE.
  305 IF (.NOT.TAUFLG) GO TO 998
      M = 0
  535 316 CONTINUE
      M = M + 1
      IF (VARPAR.LE.2) GO TO 555
      CALL SURVT(LAMDA(J),TAU(J),U(1,J),V(1,J),H(1,J),LEAKGE(1,J)
     *          ,T,SURVIV(1,J),NR(J),NT)
  540 IF (VARPAR.GE.4.AND.VARPAR.LE.20) GO TO 557
      GO TO 556
  557 CONTINUE
      CALL FLTCR(LAMDA(4),TAU(4),U(1,4),V(1,4),H(1,4),LEAKGE(1,4),
     *      T,SURVIV(1,4),NT,NR(4),LAMDA(24),LAMDA(25))
  545 GO TO 556
  555 CONTINUE
      CALL MCOSC(LAMDA,TAU,U,V,H,LEAKGE,T,SURVIV(1,2),NT)
  556 CONTINUE
      PARAM(M) = TAU(J)
  550 RESULT(M) = SURVIV(1,JS)
```

```
      FCBUS(M) = SURVIV(1,JF)
      TAU(VAPPAR) = TAU(VARPAR) + DELTAU
      IF (TAU(VARPAR).LE.TAUMAX) GO TO 316
      TAU(VARPAR) = ORGVAL(22)
555   CALL VAPPRT(PARAM,RESULT,T(1),M,UNITN(J),SURVIV(1,25),SURVIV(1,26)
     *  ,HEADER(6),6H     T,6HAU    ,2H  ,FCBUS,JF)
      TAUFLG = .FALSE.
  998 ITER = ITER - 1
      IF (ITER.GE.1)  GO TO 1
560   STOP
  999 WRITE(06,113)
  113 FORMAT(1H0,45HRUN ABORTED BECAUSE OF VARIABLES OUT OF RANGE)
      STOP
      END
```

```
      SUBROUTINE SURVT(LAMDA,TAU,U,V,H,LEAKGE,T,SURVIV,N,NT)
      REAL      SIGMA(5),LEAKGE(5),COVRGE(5),U(5),V(5),H(5),DELTA(5),
     *          ALPHA(5,5),SUM,TEMP,
     *          LAMDA,TAU,
     *          T(20),SURVIV(20)
C
C         SIGMAT CALCULATION
C
      SIGMAT = LAMDA + TAU
C
C         SIGMA(K) CALCULATION
C
      DO 10 K=1,N,1
        SIGMA(K) = LAMDA + LEAKGE(K)*TAU
   10   CONTINUE
C
C         DELTA(K) CALCULATION
C
      DO 20 K=1,N,1
        DELTA(K) = U(K)*SIGMA(K) + (1.-U(K))*SIGMAT
   20   CONTINUE
C
C         COVERAGE CALCULATION
C
      DO 30 K=1,N,1
        COVRGE(K) = U(K)*V(K)*H(K)
   30   CONTINUE
C
C         ALPHA COEFFICIENT CALCULATION
C
      ALPHA(1,1) = 1.
      IF (N.LE.1) GO TO 45
      DO 40 J=2,N,1
       L = J-1
        DO 50 K=1,L,1
          ALPHA(J,K) = J*COVRGE(J)*SIGMA(J)*ALPHA(J-1,K)/(J*DELTA(J)
     *                    - K*DELTA(K))
   50     CONTINUE
        SUM = 0.0
          DO 60 I=1,L,1
            SUM = SUM + ALPHA(J,I)
   60       CONTINUE
        ALPHA(J,J) = 1. - SUM
   40   CONTINUE
   45 CONTINUE
C
C         SURVIVABILITY CALCULATION
C
      DO 80 M=1,NT
      TEMP=0.
      DO 70 I=1,N
      TEMP = TEMP + ALPHA(N,I)*EXP(-I*DELTA(I)*T(M))
   70 CONTINUE
      SURVIV(M)= TEMP
   80 CONTINUE
```

```
      SUBROUTINE  SURVT

                  RETURN
                  END
```

```
      SUBROUTINE HCDSC(LAMDA,TAU,U,V,W,LEAKGE,T,SURVIV,NT)
      REAL LAMDA(2),TAU(2),U(5),V(5),W(5),LEAKGE(5),C(5),
     *   SIGOT,SIGKT,SIG03,SIG02,SIG01,SIGK2,SIGK1,DLT03,
     *   DLT02,DLT01,DLTK2,DLTK1,DD201,DK2031,DK2032,DK2021,
     *   S12,S21,S22,S11,NKD,COEFA,COEFB,COEFC,COEFD,COEFE,COEFF,
     *   COEFG,COEFH,COEFI,COEFJ,COEFK,COEFL,COEFM,COEFN,COEFP,
     *   TEMP,S23,T(20),SURVIV(20),ND2,ND3,NK2
C
C     CALCULATE COVERAGE
C
      DO  10 I=1,5
      C(I)=U(I)*V(I)*W(I)
   10 CONTINUE
C
C     CALCULATE SIGMAS
C
      SIGOT = LAMDA(1) + TAU(1)
      SIGKT = LAMDA(2) + TAU(2)
      SIG03 = LAMDA(1) + LEAKGE(3)*TAU(1)
      SIG02 = LAMDA(1) + LEAKGE(2)*TAU(1)
      SIG01 = LAMDA(1) + LEAKGE(1)*TAU(1)
      SIGK2 = LAMDA(2) + LEAKGE(5)*TAU(2)
      SIGK1 = LAMDA(2) + LEAKGE(4)*TAU(2)
C
C     CALCULATE DELTAS
C
      DLT03 = SIG03 * U(3) + (1-U(3))* SIGOT
      DLT02 = SIG02 * U(2) + (1-U(2))* SIGOT
      DLT01 = SIG01 * U(1) + (1-U(1))* SIGOT
      DLTK2 = SIGK2 * U(5) + (1-U(5))* SIGKT
      DLTK1 = SIGK1 * U(4) + (1-U(4))* SIGKT
C
C     COMPUTE DENOMINATORS
C
      DD201 = 2*DLT02-DLT01
      DK2031 =2*DLTK2+3*DLT03-DLTK1-DLT01
      DK2032 =DK2031+DLT01-2*DLT02
      DK2021 =2*DLTK2+2*DLT02-DLTK1-DLT01
      DD302  =3*DLT03-2*DLT02
      DK2K1 = 2*DLTK2-DLTK1
C
C     COMPUTE PARTIAL NUMERATORS
C
      ND3= C(3)*SIG03
      ND2= C(2)*SIG02
      NK2= C(5)*SIGK2
      NKD=4*(ND2+NK2)
C
C     COMPUTE NON-TIME VARYING COEFFICIENTS
C
      COEFA = 4*NK2*ND2/(DD201*DK2031)
      COEFB = (1-2*ND2/DD201)*2*NK2/DK2032
      COEFC = ND3*NKD/(DK2021*DK2031)
      COEFD = (3-NKD/DK2021)*ND3/DD302
      TEMP = 4*ND3*NK2*ND2/(DD201*DK2021)
```

```
                COEFF = TEMP/DK2031
                COEFG = TEMP/DD302
                TEMP = (1-2*ND2/DD201)*2*ND3*NK2/DK2K1
                COEFH = TEMP/DK2032
60              COEFI = TEMP/DD302
                TEMP = 4*ND3*ND2*NK2/(DK2K1*DK2021)
                COEFL = TEMP/DK2031
                COEFM = TEMP/DD302
                TEMP = (1-2*NK2/DK2K1)*2*ND3*ND2/DD201
65              COEFN = TEMP/(3*DLTD3-DLTD1)
                COEFP = TEMP/DD302
          C
          C     COMPUTE SIMPLEX SURVIVABILITIES
          C
70              DO 20 I=1,NT
                S23    = EXP(-(3*DLTD3+2*DLTK2)*T(I))
                S22    =  EXP(-(2*DLTD2+2*DLTK2)*T(I))
                S21    =  EXP(-(DLTD1+2*DLTK2)*T(I))
                S12    =  EXP(-(2*DLTD2+DLTK1)*T(I))
75              S11    =  EXP(-(DLTD1+DLTK1)*T(I))
          C
          C     COMPUTE SURVIVABILITY
          C
                SURVIV(I)=(COEFA+COEFC+COEFF+COEFL)*(S11-S23)+(COEFN)*(S21-S23)
80             *+(COEFB+COEFH)*(S12-S23)+ S23
               *+(COEFD      -COEFG-COEFI      -COEFM-COEFP)*(S22-S23)
             20 CONTINUE
                RETURN
                END
```

```
      SUBROUTINE FLTGR(LAMDA,TAU,U,V,H,LEAKGE,T,SURVIV,NT,NR,LA,LF)
      REAL     LAMDA(17),TAU(17),U(5,17),V(5,17),H(5,17),LEAKGE(5,17),
     *         C(5,17),SURVIV(20,26),T(20),DSUR(20,17),CEQFF(4),
     *CEQFA(4),CEQD,SSUR(20,17),RSIN(20,12),S1,TEMP2,TEMP3,S3,S2,LA,LF
      INTEGER  NR(17)
      DO 10 J=1,17
      DO  5 I=1,4
    5 C(I,J) = U(I,J)*V(I,J)*H(I,J)
      I = NR(J) + 1
      CALL SURVT (LAMDA(J),TAU(J),U(1,J),V(1,J),H(1,J),LEAKGE(1,J),
     *          T,DSUR(1,J),I,NT)
   10 CONTINUE
      DO 20 J=1,12
      I = NR(J) - 2
      CALL SURVT (LAMDA(J),TAU(J),U(1,J),V(1,J),H(1,J),LEAKGE(1,J),
     *          T,SSUR(1,J),I,NT)
      I = 1
      CALL SURVT (LAMDA(J),TAU(J),U(1,J),V(1,J),H(1,J),LEAKGE(1,J),
     *          T,RSIN(1,J),I,NT)
   20 CONTINUE
      DO 60  K=1,NT
      CEQD = C(2,13)
      DO 30 I= 1,4
      CEQFF(I) = C(I,1)
      CEQFA(I) = C(I,10)
   30 CONTINUE
      S1=LAMDA(1 )
      TEMP2=EXP(-S1*T(K))
      TEMP3=1-TEMP2
      DO 70 J=2,17
      IF (J.NE.10.AND.J.NE.13) GO TO 65
      IF (J.EQ.10) GO TO 64
      S1=LAMDA(13)
      TEMP2=EXP(-S1*T(K))
      TEMP3=1-TEMP2
      GO TO 65
   64 CONTINUE
      S1=LAMDA(10)
      TEMP2=EXP(-S1*T(K))
      TEMP3=1-TEMP2
   65 C(5,J)=0
      IF (TEMP3.EQ.0) GO TO 70
      C(5,J)=(LAMDA(J)*TEMP3-S1*TEMP2*(1- EXP (-LAMDA(J)*T(K))))/
     *          ((LAMDA(J)+S1)*TEMP3)
   70 CONTINUE
      DO 80 I=14,17
      CEQD    =CEQD    *(1-C(5,I)+C(5,I)*C(2,I))
   80 CONTINUE
      DO 90 J=2,4
      DO 95 I=2,9
      CEQFF(J)=CEQFF(J)*(1-C(5,I)+C(5,I)*C(J,I))
   95 CONTINUE
      DO 97 I=11,12
      CEQFA(J)=CEQFA(J)*(1-C(5,I)+C(5,I)*C(J,I))
   97 CONTINUE
```

```
      90 CONTINUE
         S2 = 1
         TEMP2 = 1
         DO 40 I=14,17
60       S2    = S2 * SURVIV(K,I)
         TEMP2 = TEMP2*DSUR(K,I)
      40 CONTINUE
         TEMP3 = EXP(-LAMDA(13)*T(K))
         SURVIV(K,24)=TEMP3*TEMP3*S2 + 2*CEQD*TEMP3*(1-TEMP3)*TEMP2
65       S3 = 1
         S2 = 1
         S1 = 1
         DO 50 I=3,9
         S3 = S3*SURVIV(K,I)
70       S2 = S2*DSUR(K,I)
         S1 = S1*SSUR(K,I)
      50 CONTINUE
         TEMP2 = EXP(-LAMDA(1)*T(K))
         TEMP3 = EXP(-LF*T(K))
75       SURVIV(K,25)=TEMP2*TEMP2*TEMP2*S3*(TEMP3*SURVIV(K,2)+CEQFF(4)*
     *   (1-TEMP3)*DSUR(K,2)) +3*CEQFF(3)*TEMP2*TEMP2*(1-TEMP2)*(TEMP3*
     *   DSUR(K,2)+CEQFF(4)*(1-TEMP3)*SSUR(K,2))+ 3*CEQFF(3)*CEQFF(2)*
     *   TEMP2*(1-TEMP2)**2*S1*(2*TEMP3*SSUR(K,2)+CEQFF(4)*(1-TEMP3)*
     *   RSIH(K,2) )
80       TEMP2 = EXP(-LAMDA(10)*T(K))
         TEMP3 = EXP(-LA*T(K))
         SURVIV(K,21) = TEMP2*TEMP2*TEMP2*DSUR(K,12)*(TEMP3*SURVIV(K,11)+
     *    3*CEQFA(4)*(1-TEMP3)*DSUR(K,11)) + 3*CEQFA(3)*TEMP2*TEMP2*
     *    DSUR(K,12)*(1-TEMP2)*(TEMP3*DSUR(K,11)+CEQFA(4)*(1-TEMP3)*
85       *    SSUR(K,11)) + 3*CEQFA(3)*CEQFA(2)*TEMP2*(1-TEMP2)**2*SSUR(K,12)
     *    *(2*TEMP3 *SSUR(K,11)+CEQFA(4)*(1-TEMP3)*RSIH(K,11))
      60 CONTINUE
         RETURN
         END
```

```
      SUBROUTINE SVTPRT(N,LAMDA,TAU,U,V,H,T,SURVIV,NR,LEAKGE,UNITN)
      REAL       U(5,25),V(5,25),H(5,25),LEAKGE(5,25),LAMDA(25),
     * TAU(25),T(20), SURVIV(20,28),FT
      INTEGER N,NR(25),UNITN(28)
C
C    BASELINE RESULTS PRINT ROUTINE
C
      WRITE(06,11)
   11 FORMAT(1H1)
      WRITE(06,12)
   12 FORMAT(1H0,19X,25HCONFIGURATION PARTICULARS)
      WRITE (6,13)
   13 FORMAT(1H0,16X,4HUNIT,19X,9HPERMANENT,22X,9HTRANSIENT)
      WRITE (6,14)
   14 FORMAT(1H ,16X,6HNAME  ,19X,4HRATE,27X,4HRATE)
      WRITE (6,17)
      DO 16 I=1,23
      WRITE (6,15) UNITN(I),LAMDA(I),TAU(I)
   15 FORMAT(1H ,15X,A6,09X,E20.7,11XE20.7)
   16 CONTINUE
   17 FORMAT(1H0)
      WRITE (6,17)
      WRITE (6,18)
   18 FORMAT(1H0,20X,9HSUBSCRIPT,9X,13HDETECTABILITY,5X,
     *  14HDIAGNOSABILITY,5X,14HRECOVERABILITY,8X,7HLEAKAGE)
      WRITE (6,17)
      DO 20 J=1,23
      K = NR(J)
      DO 20 I=2,K
      WRITE(6,19)  UNITN(J),I,U(I,J),V(I,J),H(I,J),LEAKGE(I,J)
   19 FORMAT(1H ,15X,A6,10X,I1,5X,4F19.7)
   20 CONTINUE
      UNITN(2) = 6HMCDS
      IF (N.GE.2) GO TO 50
      WRITE (6,11)
      WRITE (6,37) T(1)
   37 FORMAT(1H0,21X,15HMISSION TIME IS , E14.7,6H HOURS)
      WRITE (6,21)
   21 FORMAT(1H0,23X, 4HUNIT,24X,8HBASELINE,24X, 7HFAILURE)
      WRITE (6,22)
   22 FORMAT(1H ,23X,6HNAME  ,19X,13HSURVIVABILITY,20X,11HPROBABILITY)
      WRITE (6,17)
      DO 31 I=2,28
      FT = 1-SURVIV(1,I)
      WRITE (6,30) UNITN(I),SURVIV(1,I),FT
   30 FORMAT(1H0,21X,A6,18X,F14.7,20X,E14.7)
   31 CONTINUE
      RETURN
   50 DO 60 J=2,28
      WRITE (6,11)
      WRITE (6,17)
      WRITE (6,51) UNITN(J)
   51 FORMAT(1H0,15X,16HSURVIVABILITY FOR ,A6)
      WRITE (6,53)
   53 FORMAT(1H0,19X,7HMISSION,15X,13HCONFIGURATION,14X,7HFAILURE)
```

```
      WRITE (6,54)
   54 FORMAT(1H ,17X,11HTIME(HOURS),13X,13HSURVIVABILITY,12X,
     * 11HPROBABILITY)
      WRITE (6,17)
60    DO 60 I=1,M
      FT = 1- SURVIV(I,J)
      WRITE (6,52)   T(I),SURVIV(I,J), FT
   60 CONTINUE
   52 FORMAT(1H ,2F26.6,E26.6)
65    RETURN
      END
```

```
      SUBROUTINE VATPRT(N,LAMDA,TAU,U,V,W,T,SURVIV,SAFETY,MISSON,NT,
     *     VARPAR,LEAKGE,JS,JF)
      REAL U(5), V(5), W(5), LAMDA, TAU, T(20), SURVIV(20,20)
     *     ,SAFETY(20),MISSON(20),TEMP
     *   ,LEAKGE(5),FT
      INTEGER   N,VARPAR
C
C       PRINT ROUTIN FOR PARAMETER VARIATION
C       WITH TIME AS THE INDEPENDENT VARIABLE
C
      WRITE(06,11)
   11 FORMAT(1H1)
      WRITE(06,12)     VARPAR
   12 FORMAT(1HO,19X,30HCONFIGURATION PARTICULARS FOR ,A6)
      WRITE (6,23)
      WRITE(06,14) LAMDA
   14 FORMAT(1H ,14X,20HPERMANENT FAULT RATE,E20.7)
      WRITE(06,15) TAU
   15 FORMAT(1H ,14X,20HTRANSIENT FAULT RATE,E20.7)
      WRITE(06,16) (U(I),I=1,N)
   16 FORMAT(1H ,14X,14HDETECTABILITY ,5F14.7)
      WRITE(06,17) (V(I),I=1,N)
   17 FORMAT(1H ,14X,14HDIAGNOSABILITY,5F14.7)
      WRITE(06,18) (W(I),I=1,N)
   18 FORMAT(1H ,14X,14HRECOVERABILITY,5F14.7)
      WRITE (6,27) (LEAKGE(I),I=1,N)
   27 FORMAT(1H ,14X,7HLEAKAGE,7X,5F14.7)
      WRITE(06,19)
   19 FORMAT(1HO,19X,7HMISSION,15X,13HCONFIGURATION,15X,7HFAILURE )
      WRITE(06,21)
   21 FORMAT(1H ,17X,11HTIME(HOURS),13X,13HSURVIVABILITY,12X,
     * 11HPROBABILITY)
      WRITE(06,23)
   23 FORMAT(1HO)
      DO 60 I=1,NT
      FT = 1- SURVIV(I,JS)
      WRITE (6, 52) T(I), SURVIV(I,JS), FT
   60 CONTINUE
   52 FORMAT(1H ,2F26.6,E26.6)
      WRITE (6,23)
      WRITE (6,23)
      WRITE (6,61)
   61 FORMAT(1HO,15X,29HSAFETY CRITICAL SURVIVABILITY )
      WRITE(06,23)
      WRITE(06,19)
      WRITE(06,21)
      WRITE(06,23)
      DO 70 I=1,NT
      TEMP = SURVIV(I, JF) * SAFETY(I)
      FT = 1 - TEMP
      WRITE (6,52)   T(I),TEMP,FT
   70 CONTINUE
      WRITE (6,23)
      WRITE (6,23)
      WRITE (6,71)
```

```
      71 FORMAT(1H0,15X,30HMISSION CRITICAL SURVIVABILITY )
         WRITE(06,23)
         WPITE(06,19)
         WRITE(06,21)
60       WRITE(06,23)
         DO 80 I=1,NT
         TEMP = SURVIV(I, JF) * MISSON(I)
         FT = 1 - TEMP
         WRITE (6,52)   T(I),TEMP,FT
65    80 CONTINUE
         IF (JF .LT. 24) RETURN
         WRITE (6, 23)
         WRITE (6, 23)
         WRITE (6, 81)
70    81 FORMAT (1H0,15X,31HOVERALL PARTITION SURVIVABILITY)
         WRITE (6, 23)
         WRITE (6, 19)
         WRITE (6, 21)
         WRITE (6, 23)
75       DO 82  I = 1, NT
         FT = 1.0 - SURVIV(I,JF)
      82 WRITE (6, 52) T(I), SURVIV(I, JF), FT
         RETURN
         END
```

```
      SUBROUTINE VAPPRT(T,SURVIV,TVAL,M,VARPAR,SAFETY,MISSON,
     *   HEADER,HEADA,HEADB,HEADC,FCBUS,JF)
      REAL  T(20), SURVIV(20), TVAL, SAFETY, MISSON, FT, TEMP, FCBUS(20)
      INTEGER N,M,HEADER,HEADA,HEADB,HEADC,VARPAR
C
C       PRINT ROUTIN FOR PARAMETER VARIATION
C       WITH PARAMETER AS THE INDEPENDENT VARIABLE
C
      WRITE(06,11)
   11 FORMAT(1H1)
      WRITE(06,12) VARPAR
   12 FORMAT(1H0,12X,16HVARYING UNIT IS ,A6)
      WRITE(06,23)
      WRITE(06,20) TVAL
   20 FORMAT(1H0,15X,16HMISSION TIME IS ,E20.7,6H HOURS)
      WRITE(06,23)
      WRITE(06,19) HEADA,HEADB,HEADC
   19 FORMAT(1H0,16X,2A6,A2,14X,13HCONFIGURATION,19X,7HFAILURE)
      WRITE(06,21) HEADER
   21 FORMAT(1H ,20X,A6,10X,13HSURVIVABILITY,17X,11HPROBABILITY)
      WRITE(06,23)
   23 FORMAT(1H0)
      DO 25 I=1,M,1
      FT = 1- SURVIV(I)
   25 WRITE(06,24)  T(I),SURVIV(I),FT
   24 FORMAT(1H ,F26.6,15X,F14.7,15X,E14.7)
      WRITE (6,23)
      WRITE (6,23)
      WRITE (6,61)
   61 FORMAT(1H0,15X,29HSAFETY CRITICAL SURVIVABILITY )
      WRITE(06,23)
      WRITE(06,19) HEADA,HEADB,HEADC
      WRITE(06,21) HEADER
      WRITE(06,23)
      DO 30 I=1,M
      TEMP = SURVIV(I)*SAFETY
      IF (JF .GE. 24) TEMP = FCBUS(I) * SAFETY
      FT = 1 -TEMP
      WRITE (6,24) T(I), TEMP,FT
   30 CONTINUE
      WRITE (6,23)
      WRITE (6,23)
      WRITE (6,71)
   71 FORMAT(1H0,15X,30HMISSION CRITICAL SURVIVABILITY )
      WRITE(06,23)
      WRITE(06,19) HEADA,HEADB,HEADC
      WRITE(06,21) HEADER
      WRITE(06,23)
      DO 40 I=1,M
      TEMP = SURVIV(I)*MISSON
      IF (JF .GE. 24) TEMP = FCBUS(I) * MISSON
      FT = 1 -TEMP
      WRITE (6,24) T(I), TEMP,FT
   40 CONTINUE
      IF (JF .LT. 24) RETURN
```

```
      WRITE (6, 23)
      WRITE (6, 23)
      WRITE (6, 81)
   81 FORMAT (1H0,15X,31HOVERALL PARTITION SURVIVABILITY)
60    WRITE (6, 23)
      WRITE (6, 19) HEADA, HEADB, HEADC
      WRITE (6, 21) HEADER
      DO 82  I = 1, M
      FT = 1.0 - FCBUS(I)
65    82 WRITE (6, 24) T(I), FCBUS(I), FT
      RETURN
      END
```

THIS PAGE INTENTIONALLY LEFT BLANK

## B.1 SIMULATOR INPUT DECK SETUP

The input deck setup is dependent upon the system configuration. Since it is sometimes necssary to simulate the GPC partition without the FCB partition or vice versa, the simulator allows the combined simulation of both partitions and the separate simulation of either partition. The simulator input deck setup is dependent upon the partitions being simulated.

The first card of the simulator input deck specifies the type of simulation run being made and the seed for the random number generator. SIMTYP specifies the type of simulation. If SIMTYP=1, then the deck setup in Figure B.1-1 is used. The deck setup in Figure B.1-2 is used if SIMTYP=2. SIMTYP is specified in column 10 of the input card. SEED, the random number seed is specified in columns 11-20 of the card. It specifies the initial generative value for the random number generator. Multiple configurations can be simulated in one run. The input deck for each configuration is set up as described below. These input decks are combined into one large deck. A card with -1 in columns 9-10 is placed at the back of the deck to indicate no more configurations are to be simulated.

The input deck consists of several major groups of cards. Group 1 specifies the configuration of the GPC partition, group 2 specified the fault environment for the GPC partition, group 3 specifies the configuration of the FCB partition, and group 4 specifies the failure rates for the FCB partition. Group 2 is specified once for the CPU, once for the IOP, and once for the memory. Group 4 is specified once for permanent faults and once for transient faults. The detailed deck setup for each of these groups of cards are described in Sections B.1.1 through B.1.4.

Figure B.1-1 and Figure B.1-2 show the two possible input deck setups, which are dependent on the parameter SIMTYP. For the simulation that includes the GPC partition (SIMTYP=1), Group 3 and Group 4 will not be included if NMODU, which is specified on card of Group 1, is less than four. The FCB only simulation (SIMTYP=2) has several cards identified by "simulation descriptors." This card specifies the number of missions in columns 1-10 and the number of faults for mission in columns 11-20. As many of these cards can be used as desired, and then the deck is terminated by a card with a -1 in column 9-10.

The deck setup for the baseline configuration is listed in Section B.1.5. It consists of a total of 116 cards. Note that some of the cards are blank. In this case, the parameters that the cards specify are assumed to be zero.

| IDENTIFICATION | DESCRIPTION |
|---|---|
| HEADER | Specifies SIMTYP=1 and sets initial random seed |
| GROUP 1 | GPC configuration description |
| GROUP 3 | FCB Configuration description (only if NMODU=4) |
| GROUP 2 | Failure rates for CPU |
| GROUP 2 | Failure rates for IOP |
| GROUP 2 | Failure rates for memory |
| GROUP 4 | FCB component permanent fault rates (only if NMODU=4) |
| GROUP 4 | FCB component transient fault rates (only if NMODU=4) |

FIGURE B.1-1 DECK SETUP OF SIMTYP=1

| IDENTIFICATION | DESCRIPTION |
|---|---|
| HEADER | Specifies SIMTYP=2 and sets initial random seed |
| GROUP 3 | FCB configuration description |
| GROUP 4 | FCB component permanent fault rates |
| GROUP 4 | FCB component transient fault rates |
| Simulation Descriptor | Mission time and number of faults/mission |
| . | |
| . | |
| . | |
| Simulation Descriptor | Mission time and number of faults/mission |
| -1 | End of simulation descriptors |

FIGURE B.1-2 DECK SETUP FOR SIMTYP=2

B.1.1 INPUT PARAMETERS - GROUP 1

| CARD 1 | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|
| | 1-10 | NMIS1 | Number of missions to be simulated | INTEGER |
| | 10-20 | IDEBUG | Debugging specification | INTEGER |

NMIS1 — If this number is negative, it indicates an end of file to the program and the simulation is aborted. It is better however to use SIMTYP as the end of file indicator. Thus several simulations can be run at once.

IDEBUG — The debugging option was created to deal with specific difficulties met during the programming phase. Currently, if IDEBUG=1, the cause and time of each system failure is printed and the cause and time of the first 50 state transitions is printed. IDEBUG=0 specifies that no debugging is to take place. The user by modifying the program may specify other meanings of IDEBUG.

CARD 2

| | | | | |
|---|---|---|---|---|
| | 1-10 | NBOUC | Repetition period | INTEGER |
| | 11-20 | RMISTH | Mission duration (Hours) | REAL |

NBOUC — Faults (permanent and transients) are generated prior to simulation and listed sequentially in a fault table (TABLE(300,4)). However, because hundreds of thousands of faults are simulated, it is not feasible to generate all faults at once. On the other hand, it would be quite inefficient to generate them for each mission. NBOUC represents the number of missions for which faults are generated at one time. It should be chosen so that TABLE is efficiently utilized. TABLE contains at most 150 permanents and 150 transients. Thus if the whole system has a transient failure rate of 2400 per million hours, a permanent rate of 2000 per million hours and if the mission is 10 hours long, an average of 120 transients and 100 permanents are generated every 5000 missions. Thus NBOUC might be chosen as 5000. If NBOUC is too large, faults are generated which cannot be stored. A diagnostic is output and the simulation aborts.

RMISTH — The mission time is expressed in hours.

CARD 3

| | | | | |
|---|---|---|---|---|
| | 1-10 | NMODU | Number of modules | INTEGER |
| | 11-20 | MODSIM | Total number of computers | INTEGER |

| | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|

NMODU — Faults are generated for modules one thru NMODU.
MODULE 1 is the Central Processing Unit (CPU).
MODULE 2 is the Input/Output Processor (IOP).
MODULE 3 is the memory.
MODULE 4 is the Flight Critical Bus Partition (FCB).
Thus if NMODU is 3, faults are generated for only the CPU, IOP, and memory; the FCB would be ignored. If NMODU=4, the whole system is simulated.

MODSIM — MODSIM indicates the total number of computers. It must be less than or equal to 5 and greater than zero.

CARD 4

| | | | |
|---|---|---|---|
| 1-10 | NSPA | Number of spare computers | INTEGER |
| 11-20 | CONDIT | Space conditioning time | REAL |

NSPA — Number of spare computers (should be 0 if MODSIM <3). The IOPs should be nondedicated to the computers. For the Shuttle simulation NSPA should be set to zero.

CONDIT — CONDIT represents the time in milliseconds it takes to condition a spare computer. If NSPA is zero, CONDIT is ignored.

CARD 5

| | | | |
|---|---|---|---|
| 1-10 | NONDED | Dedicated/Nondedicated IOPs | INTEGER |
| 11-20 | NIO | Number of IOPs | INTEGER |

NONDED — 1: Dedicated IOPs
2: Nondedicated IOPs
For the Shuttle simulation, NONDED is 1.

NIO — Number of IOPs. This parameter is significant only if NONDED is equal to 2.

CARD 6

| | | | |
|---|---|---|---|
| 1-10 | PCOM | Impact of EEM fault on computer only | REAL |
| 11-20 | PBU | Impact of EEM fault on bus only | REAL |

PCOM

PBU — These two numbers represent the probability that an IOP fault impacts the computer or the bus but not both. If the IOPs are dedicated (e.g. the Shuttle DPS), the program computers PBUCO (the probability that a fault disables both the bus and computer) as 1-PCOM-PBU. Thus if PBU is zero all

| COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|

faults disable the associated computer. PCOM and PBU are set to zero by the program if the IOPs are nondedicated, which assumes that the computers and IOPs are fault independent.

CARD 7

| | | | |
|---|---|---|---|
| 1-10 | IROLLA | Rollahead indicator | INTEGER |
| 11-20 | RECOV | Rollahead recovery duration | REAL |
| 21-30 | MININT | Rollahead recurrence interval | REAL |
| 31-40 | RACPU | Rollahead effectiveness | REAL |

IROLLA

1: Rollahead is the recovery procedure in multiplex operation
0: No rollahead.

RECOV

RECOV represents the time in milliseconds required to complete state vector transfer and continue normal operation.

MININT

Specifies a time interval in milliseconds. If a fault occurs after a rollahead and before this time interval has elapsed it is in the same location as the previous fault. It is assumed to be a reoccurrence of the previous fault. The fault is thus assumed to be permanent, and a new rollahead is not initiated.

Probability that a transient fault, which does not cause program memory damage, is recovered from because of a rollahead, without any degradation. RACPU is normally unity assuming that all necessary information is included in the state vector transfer operation.

CARD 8

| | | | |
|---|---|---|---|
| 1-10 | IDLYRC | Delay reconfiguration | INTEGER |
| 11-20 | RCDUR | Recovery duration | REAL |
| 21-30 | RCCINT | Recurrence interval | REAL |
| 31-40 | DLYCPU | Effectiveness | REAL |

IDLYRC

1: This recovery technique is used instead of rollahead/rollback.
0: This recovery technique is not used.
The recovery action is delayed until the second detection of a fault. Thus transients, that do not result in any permanent damage to data or programs, do not cause system degradation.

RCDUR

RCDUR is the time in milliseconds required for this recovery procedure. It thus represents the overhead caused by the occurrence of a fault.

| COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|

RCCINT — RCCINT is the duration in milliseconds for which another detection of a fault in a computer will cause a system degradation. For example, if it is known that most transient faults should become inactive within two minor cycles and the minor cycle duration is 40 milliseconds, RCCINT would be 80 milliseconds.

DLYCPU — The probability that a transient fault which does not cause program memory damage is corrected by this recovery procedure.

CARD 9

| | | | |
|---|---|---|---|
| 1-10 | IROLLB | Rollback indicator | INTEGER |
| 11-20 | MAXRLB | Maximum number of consecutive roll-backs | INTEGER |
| 21-30 | RBCPU | Rollback effectiveness | REAL |

IROLLB — 0: No rollback.
1: Rollback. Rollback is a transient recovery procedure in which the occurrence of a fault causes the program segment in which it occurred to be re-executed. Suppose a transient fault occurs during the execution of a program and corrupts a calculation which is detected at a comparison point. The computer would then "rollback" to the previous comparison point, using the old data, and re-execute the program segment in which the fault occurred. If the fault was a transient and has disappeared then the program segment will execute properly and system operation will continue without degradation. In order to keep sync, all computers must "rollback", even the ones in which no fault occurred. The rollback duration is assumed to be equal to the inter-comparison time. If rollahead is specified, rollback is used only in duplex and simplex. If IDLYRC is one, rollahead and rollback are not used.

MAXRLB — MAXRLB specifies the maximum allowed number of consecutive rollbacks. Thus if the first rollback does not succeed the computer rollbacks again if MAXRLB is greater than one.

RBCPU — RBCPU is the probability that because of a rollback, a transient fault will not result in any system degradation.

CARD 10

| | | | |
|---|---|---|---|
| 1-10 | IDESCR | Memory copy indicator | INTEGER |
| 11-20 | DURMC | Memory copy duration | REAL |
| 21-30 | RMC | Memory copy recurrence interval | REAL |
| 31-40 | PSMC | Memory copy effectiveness | REAL |

| | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|

IDESCR

0: No memory copy
1: Memory copy

Memory copy is used as a secondary transient recovery procedure in multiplex computer operation if the primary recovery procedure fails. The memory of the working computer is transferred to that of the "faulty" computer. Ideally this would be done on a low-priority cycle-stealing basis by ROM BCE-programs residing in the IOPs and would not significantly affect computer performance. Upon completion of the memory copy, the state vector of a working computer would be DMA-burst transferred to the faulty computers, and all computers would continue in step. Memory transients usually result in memory damage because the memory is core destructive-read-out. Thus if a transient in a memory sense-amp results in faulty data, this faulty data is written back into memory on the write-cycle. Usually this type of fault is not corrected by other recovery procedures, but is corrected by memory copy.

DURMC

DURMC is the time in milliseconds between memory copy initiation and memory copy completion. For example, if on the average one word is transferred per millisecond, then DURMC would be 65536 milliseconds for a 64K memory (slightly more than one minute). DURMC is specified in milliseconds.

RMC

RMC is the memory copy recurrence interval in milliseconds. If a fault occurs within RMC milliseconds in a computer after recovery by memory copy in the previously faulty computer, it is assumed to be a re-detection of an earlier fault and system degradation occurs.

PSMC

PSMC is the probability that a memory copy succeeds in correcting a transient fault, so the fault doesn't result in system degradation.

CARD 11

| | | | |
|---|---|---|---|
| 1-10 | ISTART | System restart indicator | INTEGER |
| 11-20 | DURRES | Duration of system restart | REAL |

ISTART

0: No system restart.
1: System restart.

System restart is invoked upon the occurrence of faults in all computers, or in all computers except one. In this case, the system is re-initialized and the program memories are verified before normal operation continues. If the system restart lasts too long, the system fails.

DURRES

DURRES represents the duration in milliseconds of a system restart.

| CARD 12 | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|
| | 1-10 | DELAY | Delay before transient recovery | REAL |
| | 11-20 | TC | Time between | REAL |

DELAY — In order that transient faults may become inactive before transient recovery takes place, it is sometimes adviseable to delay the recovery several milliseconds. DELAY is the time in milliseconds transient recovery is delayed after detection of a fault.

TC — TC is the time between inter-computer comparisons expressed in milliseconds.

CARD 13

| | | | |
|---|---|---|---|
| 1-10 | DIAGN | Average self-test duration | REAL |
| 11-20 | DETMAX | Self-test efficacy | REAL |
| 21-30 | TW2 | Isolation duration | REAL |

DIAGN — The self test program is a software routine that is run to determine if the computer is faulty. The time required to diagnose a computer as faulty varies depending on the location of the fault. IDAGN represents the average time (in milliseconds) required to detect the fault.

DETMAX — DETMAX is the probability of detecting a fault by means of the STP program.

TW2 — TW2 is the time in milliseconds required for isolating a faulty computer when the system degrades from duplex to simplex.

CARD 14

| | | | |
|---|---|---|---|
| 1-10 | PDET | CPU fault detection probability | REAL |
| 11-20 | PDM | Memory fault detection probability | REAL |
| 21-30 | PDETIO | IOP fault detection probability | REAL |

PDET, PDM PDETIO — These parameters represent the probability of detecting a fault in the respective units by means of BITE. (The built-in test equipment.)

CARD 15

| | | | |
|---|---|---|---|
| 1-10 | RTI | Iteration period | REAL |
| 11-20 | MINCY | Minor cycle duration | REAL |
| 21-30 | MACY | Number of minor cycles/major cycle | INTEGER |
| 31-40 | DOWMAX | Maximum downtime | REAL |

| | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|

RTI — RTI is the iteration period in milliseconds for the control loop calculations.

MINCY — MINCY is the minor cycle duration in milliseconds. For example, for the Shuttle, MINCY is 40 milliseconds.

MACY — MACY is the number of minor cycles per major cycle.

DOWMAX — DOWMAX is the maximum allowable down time in milliseconds before a safety-critical failure occurs.

CARD 16

| | | | |
|---|---|---|---|
| 1-10 | PROMM | Relative size of minor cycle program | REAL |
| 11-20 | PSUC | Program survivability | REAL |

PROMM — PROMM is the quotient size of minor cycle program over the total number of memory words. It is used for estimating the time required for detection of a fault.

PSUC — PSUC represents the probability that the program survives when a memory fault occurs.

CARD 17

| | | | |
|---|---|---|---|
| 1-10 | ISYNC | Synchronous/asynchronous executive | INTEGER |
| 11-20 | RATINT | Interrupt rate | REAL |

ISYNC — 0: Asynchronous scheduling
1: Synchronous scheduling

RATINT — RATINT is the average number of interrupts that occur per second.

THIS PAGE INTENTIONALLY LEFT BLANK

B.1.2 INPUT PARAMETERS - GROUP 2

| CARD 1 | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|
| | 1 | DPER | Permanent distribution type | INTEGER |
| | 2 | DTRA | Transient distribution type | INTEGER |
| | 3 | DDUR | Duration distribution type | INTEGER |

DPER(I) "1" Specifies that permanent faults have an exponential interarrival time. (No other distributions are implemented).

DTRA(I) "1" Specifies that transient faults have an exponential interarrival time.
"2" Specifies that transient faults have a BURST distribution.

DDUR(I) "1" The duration of transient faults in Module-I is uniformly distributed.
"2" The duration is exponentially distributed.

CARD 2

| | | | | |
|---|---|---|---|---|
| | 1-10 | RLAMDP(I) | Permanent failure | REAL |

RLAMDP(I) This parameter is the permanent failure rate per million hours for MODULE-I.

CARD 2.1

| | | | | |
|---|---|---|---|---|
| (Only if NSPA≠0) | 1-10 | RMUP(I) | Spare dormant failure rate | REAL |

RMUP(I) This parameter specifies the dormant failure rate per hour of a spare computer. This card must be excluded from the input deck if NSPA=0.

CARD 3

| | | | | |
|---|---|---|---|---|
| If DTRA(I) =1 | 1-10 | RLAMBDT(I) | Transient failure rate | REAL |
| If DTRA(I) =2 | 1-10 | RLAMB(I) | Burst rate | REAL |
| | 11-20 | DURA(I) | Average duration of a burst | REAL |
| | 21-30 | BURST(I) | Transient fault rate during burst | REAL |

The format of this card is dependent upon the value of DTRA(I). If DTRA(I)=1, then RLAMBDT is specified. If DTRA(I)=2, then RLAMB(I), DURA(I), and BURST(I) are specified.

| | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|
| RLAMDT(I) | | | The transient failure occurrence rate in faults per million hours. | |
| RLAMB(I) | | | The transient burst rate in burst/million hours. | |
| DURA(I) | | | The average duration of a transient burst in seconds. | |
| BURST(I) | | | The transient fault rate during a burst. This is specified in faults/second. | |

CARD 4

| | | | | |
|---|---|---|---|---|
| If DDUR(I) =1 | 1-10 | RMINI(I) | Minimum transient duration | REAL |
| | 11-20 | RMAXI(I) | Minimum transient duration | REAL |
| If DDUR(I) =2 | 1-10 | AVDUR(I) | Average transient duration | REAL |

RMINI(I) RMINI(I) and RMAXI(I) are specified if the transient duration is modeled as being uniformally distributed. They represent the minimum and maximum durations in milliseconds of transient faults.

AVDUR(I) AVDUR(I) is specified only if the transient duration is modeled as being exponentially distributed. In this case AVDUR(I) represents the average transient duration in milliseconds.

B.1.3 INPUT PARAMETERS - GROUP 3

| CARD 1 | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|
| | 1-10 | NBTU | Number of bus terminal units | INTEGER |

NBTU — This parameter specifies the number of devices interfaced directly with the flight critical buses. For the current simulator implementation it should be specified as "10."

CARDS 2-9

| | | | | |
|---|---|---|---|---|
| I=1-8 | 1-2 | FCB(I,1) | Interface between bus -I and DDU-1 | INTEGER |
| | 3-4 | FCB(I,2) | Interface between bus -I and DDU-2 | INTEGER |
| | 5-6 | FCB(I,3) | Interface between bus -I and MDM FF-1 | INTEGER |
| | 7-8 | FCB(I,4) | Interface between bus -I and MDM FF-2 | INTEGER |
| | 9-10 | FCB(I,5) | Interface between bus -I and MDM FF-3 | INTEGER |
| | 11-12 | FCB(I,6) | Interface between bus -I and MDM FF-4 | INTEGER |
| | 13-14 | FCB(I,7) | Interface between bus -I and MDM FA-1 | INTEGER |
| | 15-16 | FCB(I,8) | Interface between bus -I and MDM FA-2 | INTEGER |
| | 17-18 | FCB(I,9) | Interface between bus -I and MDM FA-3 | INTEGER |
| | 19-20 | FCB(I,10) | Interface between bus -I and MDM FA-4 | INTEGER |

FCB — FCB is a matrix that describes the interface between the flight critical buses and the bus terminal units (MDMs and DDUs). A matrix element is defined by:

$$FCB(I,J) = \begin{cases} 0 \text{ if BUS-I is not interfaced with BTU-J} \\ 1 \text{ if BUS-I and BTU-J have an active interface} \\ 2 \text{ if BUS-I and BTU-J have a "secondary" interface} \end{cases}$$

An interface is considered to be active if the bus is connected to the BTU's primary port. An interface is secondary if the bus is connected to the BTU's backup port. Eight cards are needed to specify the FCB matrix. The interface for bus -I is defined on card I+1.

CARD 10

| | | | |
|---|---|---|---|
| 1-2 | BTUTYP(1) | BTU -1 identifier =1 | INTEGER |
| 3-4 | BTUTYP(2) | BTU -2 identifier =1 | INTEGER |
| 5-6 | BTUTYP(3) | BTU -3 identifier =2 | INTEGER |
| 7-8 | BTUTYP(4) | BTU -4 identifier =2 | INTEGER |
| 9-10 | BTUTYP(5) | BTU -5 identifier =2 | INTEGER |
| 11-12 | BTUTYP(6) | BTU -6 identifier =2 | INTEGER |
| 13-14 | BTUTYP(7) | BTU -7 identifier =3 | INTEGER |
| 15-16 | BTUTYP(8) | BTU -8 identifier =3 | INTEGER |
| 17-18 | BTUTYP(9) | BTU -9 identifier =3 | INTEGER |
| 19-20 | BTUTYP(10) | BTU -10 identifier =3 | INTEGER |

| | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|

BTUTYP — BTUTYP is a vector that specifies whether a bus terminal unit is a DDU, a FF MDM, or a FA MDM.

$$BTUTYP(I) = \begin{cases} 0 - \text{BTU I is a DDU} \\ 1 - \text{BTU I is a FF-MDM} \\ 2 - \text{BTU I is a FA-MDM} \end{cases}$$

The nominal values of BTUTYP for the baseline configuration are listed along with each element description.

CARD 11

| | | | | |
|---|---|---|---|---|
| | 1-10 | NDDUDV | Number of DDU devices | INTEGER |

NDDUDV — NDDUDV is the number of devices interfaced to the GPCs by means of the DDUs. The maximum value of NDDUDV is 4.

CARDS 12-13

| | | | | |
|---|---|---|---|---|
| I=1,2 | 1-2 | DDUDV(I,1) | DDU -I device 1 interface | INTEGER |
| | 3-4 | DDUDV(I,2) | DDU -I device 2 interface | INTEGER |
| | 5-6 | DDUDV(I,3) | DDU -I device 3 interface | INTEGER |
| | 7-8 | DDUDV(I,4) | DDU -I device 4 interface | INTEGER |

DDUDV — This matrix specifies which devices are connected to the dedicated display units.

$$DDUDV(I,J) = \begin{cases} 0 - \text{DDU -I does not control device of type -J} \\ 1 - \text{DDU -I does control device of type -J} \end{cases}$$

Two cards are required to specify DDUDV.

CARD 14

| | | | | |
|---|---|---|---|---|
| | 1-10 | NDFFDV | Number of dedicated FF-MDM devices | INTEGER |

NDFFDV — This parameter indicates the number of distinct devices dedicated to at least one of the FF MDMs. Should be less than 7.

CARDS 15-18

| | | | | |
|---|---|---|---|---|
| I=1,4 | 1-2 | DFFDV(I,1) | FF MDM I - dedicated device -1 interface | INTEGER |
| | 3-4 | DFFDV(I,2) | FF MDM I - dedicated device 2 interface | INTEGER |
| | 5-6 | DFFDV(I,3) | FF MDM I - dedicated device 3 interface | INTEGER |

| COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|
| 7-8 | DFFDV(I,4) | FF MDM I - dedicated device 4 interface | INTEGER+ |
| 9-10 | DFFDV(I,5) | FF MDM I - dedicated device 5 interface | INTEGER |
| 11-12 | DFFDV(I,6) | FF MDM I - dedicated device 6 interface | INTEGER |

DFFDV

This array specifies the devices that are dedicated to the flight forward MDMs. Each element of the array can take on a value of zero or one (see above description of DDUDV). Four cards are required to specify DFFDV. The devices interfaced with MDM FF-I are specified on card 14+I.

CARD 19

| COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|
| 1-10 | NNFADV | Number of non-dedicated FF MDM devices | INTEGER |

NNFADV

This specifies the number of non-dedicated devices interfaced with the FF MDMs. A maximum value of 4 can be specified. For the baseline system there are four devices.

CARD 20

| COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|
| 1-2 | NFFDVS(1) | Number of type 1 devices | INTEGER |
| 3-4 | NFFDVS(2) | Number of type 2 devices | INTEGER |
| 5-6 | NFFDVS(3) | Number of type 3 devices | INTEGER |
| 7-8 | NFFDVS(4) | Number of type 4 devices | INTEGER |

NFFADVS

The non-dedicated devices are assumed to be interfaced with MDMs FF1 - FF3. NFFDVS(I) indicates the number of redundant FF-MDM devices of type -I.

CARD 21

| COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|
| 1-10 | NDFADV | Number of FA MDM devices | INTEGER |

NDFADV

NDFADV is the number of devices connected to each of the flight aft MDMs. A maximum of three devices can be connected to the FA-MDMs. The baseline configuration uses two.

CARDS 22-25

| COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|
| 1-2 | DFADV(I,1) | MDM FA-I, device 1 connection | INTEGER |
| 3-4 | DFADV(I,2) | MDM FA-I, device 2 connection | INTEGER |
| 5-6 | DFADV(I,3) | MDM FA-I, device 3 connection | INTEGER |

| | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|
| DFADV | | | DFADV specifies the interconnections and redundancy of the flight critical devices interfaced to the FA MDM equipment buses. Four cards are required to specify DFADV; the devices interfaced with MDM FA -Is equipment bus is specified on card 21+I. | |
| CARD 26 | | | | |
| | 1-10 | BUSNM(1) | "FC - BUS 1" | CHAR* |
| | 11-20 | BUSNM(2) | "FC - BUS 2" | CHAR |
| | 21-30 | BUSNM(3) | "FC - BUS 3" | CHAR |
| | 31-40 | BUSNM(4) | "FC - BUS 4" | CHAR |
| | 41-50 | BUSNM(5) | "FC - BUS 5" | CHAR |
| | 51-60 | BUSNM(6) | "FC - BUS 6" | CHAR |
| | 61-70 | BUSNM(7) | "FC - BUS 7" | CHAR |
| | 71-80 | BUSNM(8) | "FC - BUS 8" | CHAR |
| BUSNM | | | BUSNM is a vector of names which are listed on printouts to identify the main bus parameters and data. For the Shuttle simulator BUSNM should be set to the names listed in the parameter description area. This parameter affects only the simulator listing. | |
| CARD 27 | | | | |
| | 1-10 | MDDUNM(1) | "DDU 1" | CHAR |
| | 11-20 | MDDUNM(2) | "DDU 2" | CHAR |
| MDDUNM | | | MDDUNM contains names for the first two BTUs. They are used to identify the DDUs on the output listing. | |
| CARD 28 | | | | |
| | 1-10 | MFFNM(1) | "MDM FF-1" | CHAR |
| | 11-20 | MFFNM(2) | "MDM FF-2" | CHAR |
| | 21-30 | MFFNM(3) | "MDM FF-3" | CHAR |
| | 31-40 | MFFNM(4) | "MDM FF-4" | CHAR |
| MFFNM | | | MFFNM contains names for identifying the flight forward MDMs on the simulator listing. | |
| CARD 29 | | | | |
| | 1-10 | MFANM(1) | "MDM FA-1" | CHAR |
| | 11-20 | MFANM(2) | "MDM FA-2" | CHAR |
| | 21-30 | MFANM(3) | "MDM FA-3" | CHAR |
| | 31-40 | MFANM(4) | "MDM FA-4" | CHAR |

*CHAR is short for character.

| | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|
| NFANM | | | This card specifies the names, to be printed on simulator listings, for the flight aft MDMs. | |
| CARD 30 | | | | |
| | 1-10 | DDUNM(1) | "AVVI" | CHAR |
| | 11-20 | DDUNM(2) | "AMI" | CHAR |
| | 21-30 | DDUNM(3) | "HSI" | CHAR |
| | 31-40 | DDUNM(4) | "ADI" | CHAR |
| DDUNM | | | DDUNM is a vector of 10 character names used to identify the devices interfaced with the DDUs on simulator output listings. The parameter description above give the names used for the baseline configuration. | |
| CARD 31 | | | | |
| | 1-10 | DFFNM(1) | "ADTA" | CHAR |
| | 11-20 | DFFNM(2) | "ACCEL" | CHAR |
| | 21-30 | DFFNM(3) | "IMU" | CHAR |
| | 31-40 | DFFNM(4) | "TACAN" | CHAR |
| | 41-50 | DFFNM(5) | "MSBLS" | CHAR |
| | 51-60 | DFFNM(6) | "RALT" | CHAR |
| DFFNM | | | DFFNM contains the identification names for the FF-MDM dedicated devices. They are used for simulator listings. | |
| CARD 32 | | | | |
| | 1-10 | NFFNM(1) | "MTU" | CHAR |
| | 11-20 | NFFNM(2) | "RHC" | CHAR |
| | 21-30 | NFFNM(3) | "RPTA" | CHAR |
| | 31-40 | NFFNM(4) | "SBTC" | CHAR |
| NFFNM | | | NFFNM contains the identifications for the non-dedicated FF-MDM devices. | |
| CARD 33 | | | | |
| | 1-10 | DFANM(1) | "ASA" | CHAR |
| | 11-20 | DFANM(2) | "RGYRO" | CHAR |
| DFANM | | | This card specifies names for identifying the devices connected to the FA-MDMs. | |

| CARD 34 | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|
| | 1-10 | BUSTFL | Bus transient fault leakage | REAL |
| | 11-20 | BUSCOV | Bus coverage | REAL |

BUSTFL — The bus transient fault leakage represents the probability that a transient fault on the bus causes unnecessary bus removal.

BUSCOV — The bus coverage is the probability that a bus fault will not result in a system failure.

CARDS 35-44

| | | | |
|---|---|---|---|
| 1-10 | BTUTFL(J,1) | Transient leakage for BTU port | REAL |
| 11-20 | BTUTFL(J,2) | Transient leakage for BTU | REAL |
| 21-30 | BTUCOV(J,1) | Coverage of BTU port fault | REAL |
| 31-40 | BTUCOV(J,2) | Coverage of fault in whole BTU | REAL |

BTUTFL — BTUTFL is a matrix of the transient fault leakages for the DDUs and MDMs. BTUTFL(J,1) is the probability that a transient fault causes system degradation given that it occurs in the redundant portion of BTU-J. BTUTFL(J,2) the transient leakage for the non redundant portion of BTU-J.

BTUCOV — BTUCOV is a matrix of fault coverages. The first column of BTUCOV contains coverages for failures in the redundant part of an MDM or DDU (i.e. the MIA, A/D, SCI, etc.). The second column specifies the coverages for the non redundant part of a BTU. Ten cards are required to specify the leakages and coverages for all the BTUs. BTU J is specified by card 84+J.

CARD 45

| | | | |
|---|---|---|---|
| 1-10 | DDUTFL(1) | Transient fault leakage for AVVI | REAL |
| 11-20 | DDUTFL(2) | Transient fault leakage for AMI | REAL |
| 21-30 | DDUTFL(3) | Transient fault leakage for HSI | REAL |
| 31-40 | DDUTFL(4) | Transient fault leakage for ADI | REAL |

DDUTFL(I) — This parameter specifies the transient fault leakage for device -I of a DDU. It should be a number between zero and one.

CARD 46

| | | | |
|---|---|---|---|
| 1-10 | DDUCOV(1) | Coverage for fault in AVVI | REAL |
| 11-20 | DDUCOV(2) | Coverage for fault in AMI | REAL |
| 21-30 | DDUCOV(3) | Coverage for fault in HSI | REAL |
| 31-40 | DDUCOV(4) | Coverage for fault in ADI | REAL |

DDUCOV — DDUCOV(I) is the coverage for faults occurring in DDU device -I. The coverage represents the probability that if a fault occurs in a DDU device and its redundant counterpart if working, then the system will recover.

| COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|
| **CARDS 47-48** | | | |
| 1-10 | DDUTFD(I,1) | AVVI transient fault detectability | REAL |
| 11-20 | DDUTFD(I,2) | AMI transient fault detectability | REAL |
| 21-30 | DDUTFD(I,3) | HSI transient fault detectability | REAL |
| 31-40 | DDUTFD(I,4) | ADI transient fault detectability | REAL |

DDUTFD — The transient fault detectability is defined as the probability that a transient fault is detected given that it occurs. Two cards are required. The first card specifies the transient detectability of a DDU device not having a redundant counterpart (i.e. its redundant counterpart has failed). The second card specifies the transient fault detectability for a device having one redundant counterpart.

| | | | |
|---|---|---|---|
| **CARD 49** | | | |
| 1-10 | DFFTFL(1) | ADTA transient fault leakage | REAL |
| 11-20 | DFFTFL(2) | ACCEL transient fault leakage | REAL |
| 21-30 | DFFTFL(3) | IMU transient fault leakage | REAL |
| 31-40 | DFFTFL(4) | TACAN transient fault leakage | REAL |
| 41-50 | DFFTFL(5) | MSBLS transient fault leakage | REAL |
| 51-60 | DFFTFL(6) | RALT transient fault leakage | REAL |

DFFTFL — This parameter specifies the transient fault leakages for each of the FF-MDM dedicated devices.

| | | | |
|---|---|---|---|
| **CARDS 50-52** | | | |
| 1-10 | DFFCOV(1,J) | ADTA fault coverage | REAL |
| 11-20 | DFFCOV(2,J) | ACCEL fault coverage | REAL |
| 21-30 | DFFCOV(3,J) | IMU fault coverage | REAL |
| 31-40 | DFFCOV(4,J) | TACAN fault coverage | REAL |
| 41-50 | DFFCOV(5,J) | MSBLS fault coverage | REAL |
| 51-60 | DFFCOV(6,J) | RALT fault coverage | REAL |

DFFCOV — This parameter specifies the permanent fault coverages for the dedicated FF-MDM devices at each of their possible redundancy levels. Three cards are required to specify DFFCOV. Card 50 lists the coverages for a fault occurring when there are two devices; Card 51 lists the coverages when there are three redundant devices; and Card 52 lists the coverages when there are four redundant devices.

| COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|
| **CARDS 53-56** | | | |
| 1-10 | DFFTFD(1,J) | ADTA transient fault detectability | REAL |
| 11-20 | DFFTFD(2,J) | ACCEL transient fault detectability | REAL |
| 21-30 | DFFTFD(3,J) | IMU transient fault detectability | REAL |
| 31-40 | DFFTFD(4,J) | TACAN transient fault detectability | REAL |
| 41-50 | DFFTFD(5,J) | MSBLS transient fault detectability | REAL |
| 51-60 | DFFTFD(6,J) | RALT transient fault detectability | REAL |

DFFTFD

This parameter lists the transient fault detection probabilities for faults occurring in the dedicated devices. Four cards are required to specify the DFFTFD array. Card 53 specifies the transient fault detectability for a device having no redundant counterparts left. Card 54, Card 55 and Card 56 specify the transient fault detectionwhen there are one, two or three redundant counterparts to the device left.

| | | | |
|---|---|---|---|
| **CARDS 57-58** | | | |
| 1-10 | NFFTFL(1,J) | MTU transient fault leakage | REAL |
| 11-20 | NFFTFL(2,J) | RMC transient fault leakage | REAL |
| 21-30 | NFFTFL(3,J) | RPTA transient fault leakage | REAL |
| 31-40 | NFFTFL(4,J) | SBTC transient fault leakage | REAL |

NFFTFL

Card 57 specifies the transient fault leakages for a transient fault affecting the whole unit. Card 58 specifies the transient fault leakages for transient faults affecting the redundant portion of the non-dedicated FF-MDM devices.

| | | | |
|---|---|---|---|
| **CARDS 59-60** | | | |
| 1-10 | NFFTFD(1,J) | Transient fault detection for MTU | REAL |
| 11-20 | NFFTFD(2,J) | Transient fault detection for RMC | REAL |
| 21-30 | NFFTFD(3,J) | Transient fault detection for RPTA | REAL |
| 31-40 | NFFTFD(4,J) | Transient fault detection for SBTC | REAL |

NFFTFD

Card 59 specifies the transient fault detection probability for faults affecting the whole unit. Card 60 specifies the transient fault detection probability for faults affecting the redundant part of each unit.

| | | | |
|---|---|---|---|
| **CARD 61** | | | |
| 1-10 | NFFPFD(1) | Permanent fault detectability for MTU | REAL |
| 11-20 | NFFPFD(2) | Permanent fault detectability for RMC | REAL |
| 21-30 | NFFPFD(3) | Permanent fault detectability for RPTA | REAL |
| 31-40 | NFFPFD(4) | Permanent fault detectability for SBTC | REAL |

NFFPFD

This card specifies the permanent fault detection probability for faults affecting the entire device.

| CARD 62 | COLUMN | PARAMETER NAME | PARAMETER DESCRIPTION | DATA TYPE |
|---|---|---|---|---|
| | 1-10 | NFFCOV(1) | MTU fault coverage | REAL |
| | 11-20 | NFFCOV(2) | RMC fault coverage | REAL |
| | 21-30 | NFFCOV(3) | RPTA fault coverage | REAL |
| | 31-40 | NFFCOV(4) | SBTC fault coverage | REAL |

NFFCOV — This card specifies the permanent fault coverage for each of the non-dedicated devices interfaced with the FF-MDMs.

CARD 63

| | | | | |
|---|---|---|---|---|
| | 1-10 | DFATFL(1) | ASA transient fault leakage | REAL |
| | 11-20 | DFATFL(2) | RGYRO transient fault leakage | REAL |

DFATFL — This card specifies the transient fault leakages for the devices interfaced to the FA-MDMs.

CARD 64

| | | | | |
|---|---|---|---|---|
| | 1-10 | DFATFD(1) | ASA transient detectability | REAL |
| | 11-20 | DFATFD(2) | RGYRO transient detectability | REAL |

DFATFD — DFATFD(1) and DFATFD(2) are the transient fault detection probabilities for the ASA and the RGYRO, respectively.

CARD 65

| | | | | |
|---|---|---|---|---|
| | 1-10 | DFACOV(1) | ASA fault coverage | REAL |
| | 11-20 | DFACOV(2) | RGYRO fault coverage | REAL |

DFACOV — This card specifies the fault isolation-recovery coverages for faults occurring in the devices linked to the flight.

THIS PAGE INTENTIONALLY LEFT BLANK

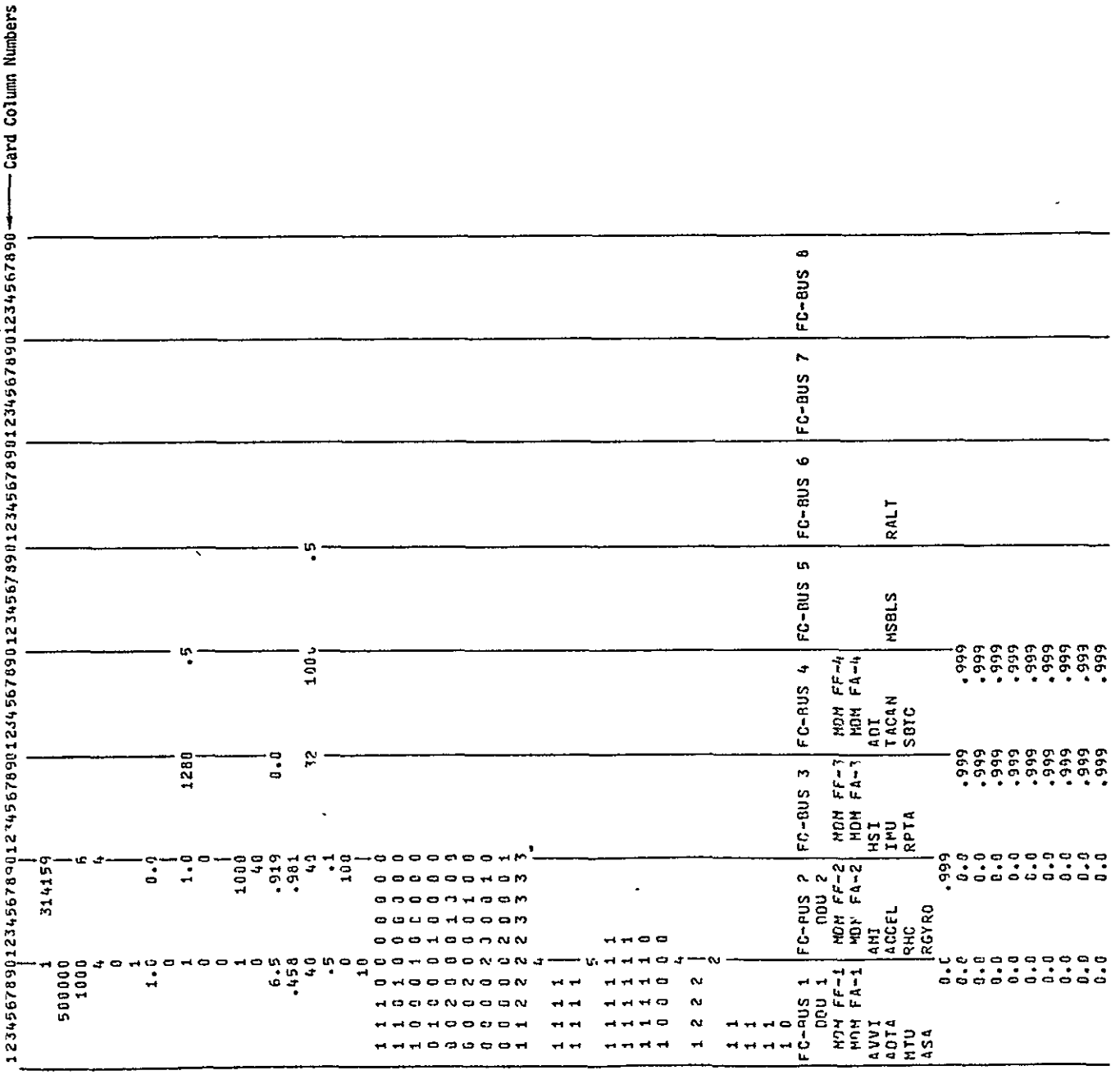
### B.1.4 INPUT PARAMETERS - GROUP 4

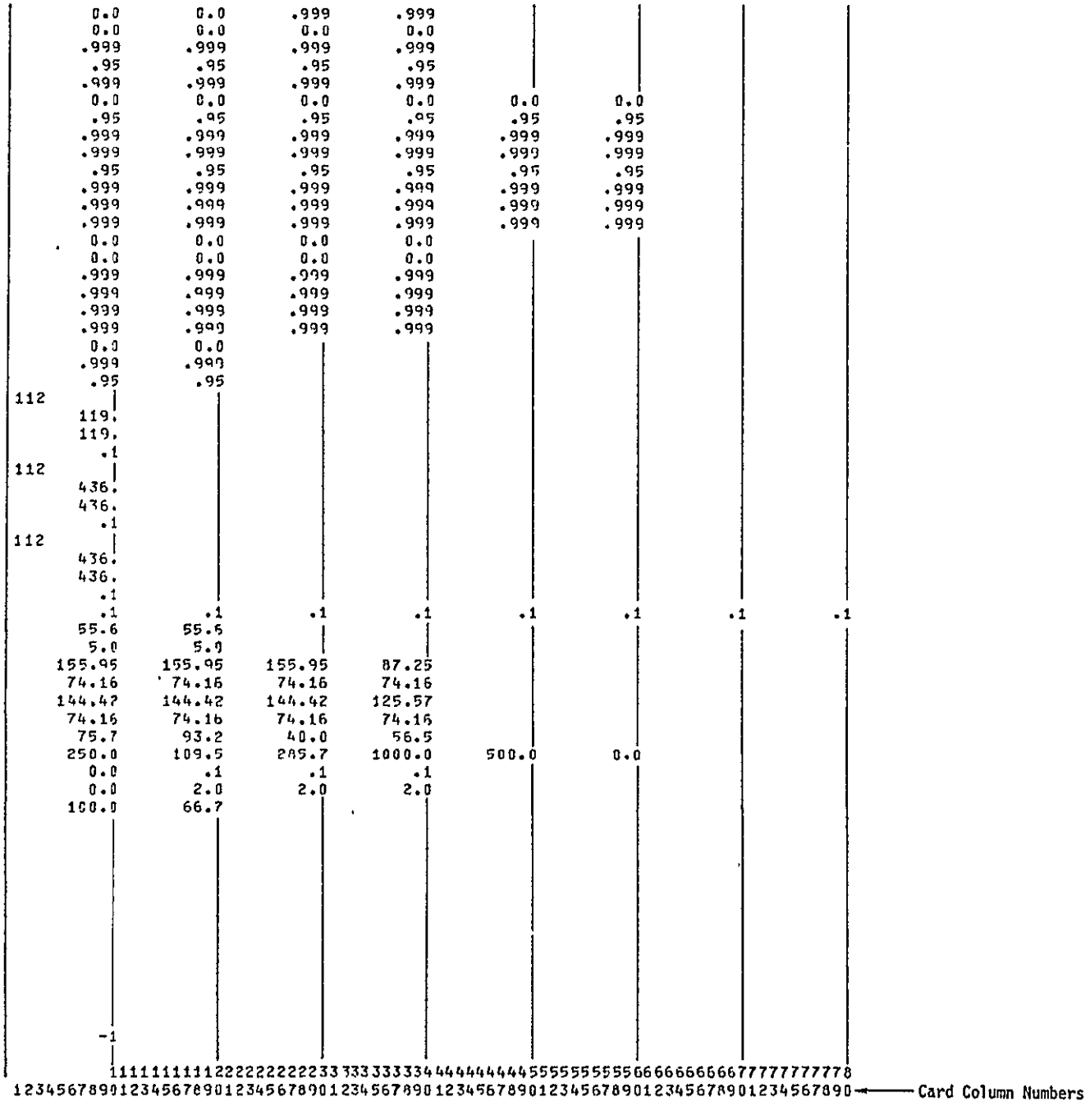
This section describes the input deck for the flight critical bus partition failure rates. The units for all failure rates in this section is faults per million hours. All inputs are real F10.0 format. The same card set-up is used for both permanent and transient failure rates.

| CARD 1 | COLUMN | PARAMETER DESCRIPTION |
|---|---|---|
| | 1-10 | Flight critical bus -1 failure rate |
| | 11-20 | Flight critical bus -2 failure rate |
| | 21-30 | Flight critical bus -3 failure rate |
| | 31-40 | Flight critical bus -4 failure rate |
| | 41-50 | Flight critical bus -5 failure rate |
| | 51-60 | Flight critical bus -6 failure rate |
| | 61-70 | Flight critical bus -7 failure rate |
| | 71-80 | Flight critical bus -8 failure rate |
| CARD 2 | | |
| | 1-10 | DDU -1 failure rate (whole unit) |
| | 11-20 | DDU -2 failure rate (whole unit) |
| CARD 3 | | |
| | 1-10 | DDU -1 failure rate (redundant portion) |
| | 11-20 | DDU -2 failure rate (redundant portion) |
| CARD 4 | | |
| | 1-10 | MDM FF-1 failure rate (whole unit) |
| | 11-20 | MDM FF-2 failure rate (whole unit) |
| | 21-30 | MDM FF-3 failure rate (whole unit) |
| | 31-40 | MDM FF-4 failure rate (whole unit) |
| CARD 5 | | |
| | 1-10 | MDM FA-1 failure rate (redundant portion) |
| | 11-20 | MDM FA-2 failure rate (redundant portion) |
| | 21-30 | MDM FA-3 failure rate (redundant portion) |
| | 31-40 | MDM FA-4 failure rate (redundant portion) |
| CARD 6 | | |
| | 1-10 | MDM FA-1 failure rate (whole unit) |
| | 11-20 | MDM FA-2 failure rate (whole unit) |
| | 21-30 | MDM FA-3 failure rate (whole unit) |
| | 31-40 | MDM FA-4 failure rate (whole unit) |

| CARD 7 | COLUMN | PARAMETER DESCRIPTION |
|---|---|---|
| | 1-10 | MDM FA-1 failure rate (redundant portion) |
| | 11-20 | MDM FA-2 failure rate (redundant portion) |
| | 21-30 | MDM FA-3 failure rate (redundant portion) |
| | 31-40 | MDM FA-4 failure rate (redundant portion) |
| CARD 8 | | |
| | 1-10 | AVVI failure rate |
| | 11-20 | AMI failure rate |
| | 21-30 | HSI failure rate |
| | 31-40 | ADI failure rate |
| CARD 9 | | |
| | 1-10 | ADTA failure rate |
| | 11-20 | ACCEL failure rate |
| | 21-30 | IMU failure rate |
| | 31-40 | TACAN failure rate |
| | 41-50 | MSBLS failure rate |
| | 51-60 | RALT failure rate |
| CARD 10 | | |
| | 1-10 | MTU failure rate (whole unit) |
| | 11-20 | RMC failure rate (whole unit) |
| | 21-30 | RPTA failure rate (whole unit) |
| | 31-40 | SBTC failure rate (whole unit) |
| CARD 11 | | |
| | 1-10 | MTU failure rate (redundant portion) |
| | 11-20 | RMC failure rate (redundant portion) |
| | 21-30 | RPTA failure rate (redundant portion) |
| | 31-40 | SBTC failure rate (redundant portion) |
| CARD 12 | | |
| | 1-10 | ASA failure rate |
| | 11-20 | RGYRO failure rate |

FIGURE B.1-5 SIMULATOR INPUT DECK FOR BASELINE CONFIGURATION




0.0 0.0 .999 .999
0.0 0.0 0.0 0.0
.999 .999 .999 .999
.95 .95 .95 .95
.999 .999 .999 .999
0.0 0.0 0.0 0.0 0.0 0.0
.95 .95 .95 .95 .95 .95
.999 .999 .999 .999 .999 .999
.999 .999 .999 .999 .999 .999
.95 .95 .95 .95 .95 .95
.999 .999 .999 .999 .999 .999
.999 .999 .999 .999 .999 .999
.999 .999 .999 .999 .999 .999
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
.999 .999 .999 .999
.999 .999 .999 .999
.999 .999 .999 .999
.999 .999 .999 .999
0.0 0.0
.999 .999
.95 .95
112
119.
119.
.1
112
436.
436.
.1
112
436.
436.
.1
.1 .1 .1 .1 .1 .1 .1 .1
55.6 55.6
5.0 5.0
155.95 155.95 155.95 87.25
74.16 74.16 74.16 74.16
144.42 144.42 144.42 125.57
74.16 74.16 74.16 74.16
75.7 93.2 40.0 56.5
250.0 109.5 285.7 1000.0 500.0 0.0
0.0 .1 .1 .1
0.0 2.0 2.0 2.0
100.0 66.7
-1
1111111111222222222233333333344444444455555555556666666666777777777 8
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890 Card Column Numbers

## B.2 SIMULATOR LISTINGS

The simulator is currently implemented as a set of fifty-eight Fortran programs designed to be compiled and run on a CDC 6600 using the extended Fortran IV compiler. Figure B.2-1 gives a short description for each of the programs. The source listings that follow this section represent the current CDC 6600 implementation of the program and will require several modifications for use in a UNIVAC 1108 computer environment.

The simulator source listings are in the same order as the subroutine names in Figure B.2-1. Some of the programs are documented by comments that describe the program, its arguments, and many of the variables that are used for the simulation. All of the listings include a comprehensive cross reference for all variables, statement labels, and subroutines. These are included in order that minor modifications to the programs can be readily made. The subroutines will be documented more thoroughly when the simulator is implemented on the UNIVAC 1108.

Several routines must be modified before the simulator can be run on the UNIVAC 1108. These routines are identified with an "*" in Figure B.2-1. PACK and UNPACK must be modified since they require non standard intrinsic functions. FCBIN, PFCBCF, and PIOSTS must be modified since they currently use a 60 bit word to represent character data. In addition, two new functions RANF and RANSET will be added for the UNIVAC 1108 implementation. These routines, which are used to generate random numbers uniformily distributed between zero and one, have already been written.

| | ROUTINE | DESCRIPTION |
|---|---|---|
| 1. | DRIVER | Main simulator program |
| 2. | ASYNC | Generate pseudo faults in case of asynchronous executive |
| 3. | BUSCHK | Simulates an IOP caused bus failure |
| 4. | BUSFLT | Simulates a bus failure |
| 5. | CLEAR | Initializes a vector to zero |
| 6. | CONFID | Generates a confidence interval for analytic model parameter estimates |
| 7. | COPY | Copies a fault into the active fault table |
| 8. | DDUFLT | Simulates faults occurring in the AVVI, AMI, HSI, and ADI |
| 9. | DETTIM | Randomly determines the detection time of a fault |
| 10. | DFAFLT | Simulates faults occurring in the ASA and RGYRO |
| 11. | DFFFLT | Simulates faults occurring in the ADTA, ACCEL, IMU, TACAN, MSBLS and RALT |
| 12. | EXDUR | Randomly generates exponential duration for transient faults |
| 13. | EXPON | Generates a table of exponentially distributed fault occurrence times |
| 14. | EXTENT | Determines if a fault causes program memory damage |
| 15. | FAUGEN | Generates faults for the simulation |
| 16. | FCBFLT | Simulates faults occurring in the flight critical bus partition |
| 17. | FCBIN | Inputs the flight critical bus partition configuration parameters |
| 18. | FCBINI | Initializes a mission for FCB simulation |
| 19. | FCBPI | Initializes the FCB partition simulator variables |
| 20. | FCBPMI | Controls simulation when SIMTYP=2 |
| 21. | FIFAU | Determines the fault location |
| 22. | GATHER | Compresses active fault table after removal of faults |
| 23. | GIORF1 | FCB failure rate input subroutine |
| 24. | GIORF2 | FCB failure rate input subroutine |
| 25. | GIORF3 | FCB failure rate input subroutine |
| 26. | GIORF4 | FCB failure rate input subroutine |
| 27. | GREATER | Randomly determines if a fault is not covered |
| 28. | INCOPY | Copies one vector to another for initialization |
| 29. | INFLTB | Routine for inputting a hand determined fault table |
| 30. | IO | Simulates faults occurring in the IOPs when they are non-dedicated |
| 31. | IRAN | Determines a random integer between 1 and X |
| 32. | ISTEPD | Chooses random integer using probability distribution vector |
| 33. | LESS | Randomly determines if a fault is covered |
| 34. | MDMFLT | Simulates the occurrence of a fault in an MDM or DDU |
| 35. | MDMPF | Simulates the removal of an MDM or DDU |
| 36. | MISCYC | Simulates the occurrence of missed iterations |
| 37. | MSTEPD | Same as ISTEPD, except row of matrix used as distribution function |
| 38. | NFFFLT | Simulates faults occurring in the MTU, RHC, RPTA, or SBTC |
| 39. | *PACK | Compresses a vector of five numbers into one number |
| 40. | *PFCBCF | Prints the FCB partition configuration definition |
| 41. | PFISO | Randomly determines where a permanent fault in the FCB partition occurs |
| 42. | PIOCNF | Prints the FCB interconnection matrices |
| 43. | *PIOSTS | Prints the FCB simulation results |
| 44. | RDIOFR | Reads FCB partition failure rates |
| 45. | SETSTS | Sets a value into the FCB status vector |
| 46. | STATE1 | Simulates normal multiplex GPC operation |
| 47. | STATE2 | Simulates a rollahead, a rollback, or delay reconfiguration in multiplex |
| 48. | STATE3 | Simulates a system restart |
| 49. | STATE4 | Simulates normal duplex GPC operation |
| 50. | STATE7 | Simulates a memory copy in multiplex |
| 51. | STATE8 | Simulates rollback or delay-reconfiguration in duplex |
| 52. | STATE9 | Simulates fault isolation in duplex |
| 53. | STATEA | Simulates simplex normal operation |
| 54. | STATEB | Simulates rollback in simplex |
| 55. | STATEC | Simulates GPC replacement by spare |
| 56. | TFISO | Randomly determines where a transient fault occurs in FCB partition |
| 57. | UNIF | Generates a uniformly distributed random number |
| 58. | *UNPACK | Undoes what PACK does |

*These routines will be modified for the UNIVAC 1108 implementation.

FIGURE B.2-1 SUMMARY OF SIMULATION ROUTINES

```
      PROGRAM DRIVER(INPUT,OUTPUT)
C                          THIS VERSION:  MARCH 1976
C ACFAU   SEE STATE1
C BUNOND  0: DEDICATED BUSSES (TO COMPUTERS)
C         1: NON-DEDICATED BUSSES
C COPLAS  LAST TIME THERE WAS A MEMORY COPY FOR COMPUTER I
C DELAY   DELAY BETWEEN DETECTION AND RECOVERY (MILLISECONDS)
C DOWMAX  MAXIMUM TOLERABLE DOWNTIME
C DURMC   DURATION OF A MEMORY-COPY (MILLISECONDS)
C DURRES SYSTEM RESTART DURATION
C ENDMIS  TIME AT WHICH THE CURRENT MISSION ENDS
C EXTEN   EXTENT OF THE LAST FAULT
C FOCO    FAULTY COMPUTER
C IDESCR  TMR DESCRIPTION  1 WITH MEMORY COPY
C                          2 WITHOUT MEMORY-COPY
C IDETEC  SEE STATE1
C IDIM    MAXIMUM NUMBER OF LURKING ERRORS                          SP09APR
C IFAU    NUMBER OF FAULTS
C IFULL   INDICATES AN OVERFLOW OF TABLE WHEN EQUAL TO 1
C ISIM    NO. OF TIMES IN SIMPLEX                                   SP09APR4
C ITLKP   TRANSIENTS MISTAKEN AS PERMANENTS IN DUPLEX
C LAST(I) LAST TIME THERE WAS A ROLLAHEAD FOR COMPUTER I (REAL)
C MACY    NUMBER OF MINOR CYCLE DURING A MAJOR CYCLE
C MINCY   MINOR CYCLE DURATION
C MININT  RECURRENCE INTERVAL (MILLISECONDS)
C MISITE  MISSED ITERATIONS
C MISTAK  TRANSIENTS MISTAKEN AS PERMANENTS
C NBOUC   NUMBER OF MISSIONS FOR WHICH THE FAULT TABLE IS DETERMINED AT
C         ONE TIME
C NDIAG   NUMBER OF DIAGNOSTICS
C NEXT = NEXT STATE. 1 : NORMAL.
C                    2 : RECOVERY ATTEMPT.
C                    3 : SYSTEM RESTART.
C                    4 : N-1 UNIT SYSTEM.
C                    5 : SYSTEM FAILURE.
C                    6 : NEXT MISSION
C                    7 : MEMORY-COPY
C                    8 : ROLLBACK
C                    9 : DIAGNOSTIC AND RECOVERY                   SP09APR4
C                   10 : SIMPLEX                                   SP09APR4
C                   11 : ROLLBACK IN SIMPLEX
C                   12 : INTRODUCTION OF A SPARE
C NFDANR  NO. SYSTEM FAILURES VIA D AND R                           SP09APR4
C NFIO    NUMBER OF FAULTS DUE TO IO FAILURES
C NFSIM   NO. SYSTEM FAILURES VIA SIMPLEX                           SP09APR4
C NMC     NUMBER OF MEMORY-COPIES
C NMIS    CURRENT MISSION
C NMIS1   TOTAL NUMBER OF MISSIONS
C NONDED  1: DEDICATED EEM'S
C         2: NON-DEDICATED EEM'S
C NOON(5) NOON(I)=1 MEANS COMPUTER I IS ON
C NWOION  NUMBER OF WORKING EEM
C NWORK   NUMBER OF WORKING COMPUTERS
C NQUA    NUMBER OF SWITCHES TO QUADRUPLEX
C NRL     NUMBER OF ROLLAHEADS
```

```
C NTR     NUMBER OF TRANSIENTS
C NTRI    NUMBER OF SWITCHES TO TRIPLEX
C NTR3    NO. OF TRANSIENTS RECOVERED FROM IN TRIPLEX                    SP08APR4
C NTR2    NO. OF TRANSIENTS RECOVERED FROM IN DUPLEX                     SP08APR4
C NUNDI   NUMBER OF UNSUCCESSFUL DIAGNOSTICS
C N2      NUMBER OF SWITCHES TO DUPLEX
C PROMM   PROPORTION OF MEMORY ALLOCATED TO MINOR CYCLE PROGRAMMING
C PSUC    PROBABILITY OF SUCCESS OF ROLL AHEAD
C PTR     POINTER TO THE FAULT TABLE
C REASON  REASON OF THE SYSTEM FAILURE
C         1: MORE THAN 10 FAULTS SHOULD BE STORED IN ACFAU. TO AVOID
C            THIS, INCREASE THE SIZE OF ACFAU AND SET IDIM TO THIS NEW
C            SIZE
C         2: FEM FAILURE
C         3: BUS OR/AND EXTERNAL DEVICE FAILURE
C         4: DUPLEX FAILURE (NON ISOLATED FAULTS)
C         5: SIMPLEX FAILURE
C         6: EXCESSIVE DOWNTIME
C RECOV   DURATION OF A ROLLAHEAD
C RMC     RECURRENCE INTERVAL FOR MEMORY COPY
C RMISTH  MISSION TIME (HOURS)
C RMISTM  MISSION TIME (MILLISECONDS)
C RTI     ITERATION PERIOD
C TABLE   SEE FAUGENE
C TC      TIME BETWEEN COMPARISONS (MILLISECONDS)
C TIME    TIME (MILLISECONDS)
C TM2     ISOLATION DURATION
C RULES OF THE GAME: AT THE END OF THE SIMULATION OF A STATE, WE HAVE:  SP09APR4
C     TIME  = TIME OF THE TRANSITION
C     PTR   = POINTER TO THE NEXT FAULT TO OCCUR
C
C ***************************************************************************
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM2/NMIS,IFULL
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/COM7/REASON
      COMMON/COM8/COCO                                                   SP09APR4
      COMMON/COM9/COPLAS(5),RMC,DURMC
      COMMON/CO10/MAXRLB,DURRB
      COMMON/CO11/MISTAK/
      COMMON/CO12/NTR,IFAU
      COMMON/CO13/ DURRES
      COMMON/CO14/RLAMDP(5),RLAMDT(5),RMINI(5),RMAXI(5),AVDUP(5)
      COMMON/CO15/NOON(5),NWORK
      COMMON/CO16/NDIAG,NUNDI
      COMMON/CO17/NTRI,NDUA
      COMMON/CO26/TM2
      COMMON/CO27/DOR,DFOR,MISTI,TISL,MI,FRTI
      COMMON/CO28/NODSIM,NSPB
      COMMON/CO29/NMC
      COMMON/CO31/RATINT,ISYNC
      COMMON/CO33/RMUP(5)
      COMMON/CO34/ISPARE(5)
      COMMON/CO35/CONDIT
      COMMON/CO36/RMISTM
```

```
      COMMON/C037/ RMU
      COMMON/C038/IOCU(5),NONDED,NHOIO
      COMMON/C039/PDETIO
      COMMON/C040/ISHI,FSHI,NSHION,NIO
115   COMMON/C041/ICATAS,I3
      COMMON/C042/MISTKI
      COMMON/C045/PSMC
      COMMON/C046/PCOM,PBU,PBUCO
      COMMON/COVER/ITLKP,IFWE                                    SP11APR4
120   COMMON/CYC/LAUNMI
      COMMON/D/DIAGN
      COMMON/DEBUG/IDEBUG
      COMMON/DETF/PDET,DETMAX,PDM
      COMMON/EXEC/MINCY,RTI,TODO,OLTIS,IDLE,SEQMAX,DOWMAX,MISITE
125   COMMON/FAILD2/FOCO2
      COMMON/FCOUNT/NF(5),NTRF(5)
      COMMON/MM/PROMM,MACY
      COMMON/PERM/LAST(5),MININT,PSUC
      COMMON/PHILM3/NTR3,NTR2,NTR1,NTNR1
130   COMMON/PAHEAD/RACPU
      COMMON/RBACK/RBCPU
      COMMON/FLTMIS/NFCV(10),NSYSF(10),PFRIO,TFRIO
      COMMON/FLTMS/IFLTCT
      INTEGER EXTEN,PTR,REASON,SEQMAX
135   INTEGER SIMTYP
      LOGICAL IOSIM
      DIMENSION NFAILI(5),MISTKI(3)
      DIMENSION NMCONF(2,5)
      REAL IDLE,LAST,LAUNMI,MINCY,MININT
140   C ***************************************************************************
      C
      DATA NMCONF/5HSTMPL,2HEX,5HOUPLE,1HX,5HTRIPL,2HEX
     * ,5HQUADR,5HUPLEX,5HQUINT,5HUPLEX/
      IDIM=10
145     10 CONTINUE
      READ 9802,SIMTYP,SEED
      IF(SIMTYP.LE.0) GOTO 20
      CALL RANSET(SEED)
      GOTO(1000,2000,3000),SIMTYP
150    1000 CONTINUE
      READ 9801,NMIS1,IDEBUG
      IF (NMIS1.LE.0) GO TO 20
      CALL CLEAR(10,NFCV)
      CALL CLEAR(10,NSYSF)
155   CALL CLEAR(5,NFAILI)
      CALL CLEAR(3,MISTKI)
      CALL CLEAR(5,NF)
      CALL CLEAR(5,NTRF)
      DOR=-1.
160   DEOR=-1.
      FOCO=1.0                                                   SP02APR4
      FOCO2=1.
      ICATAS=0
      IFAU=0
165   IFULL=0
```

```
                IOF=0
                ISIM = 0                                                SP09APR4
                ITLKP = 0                                               SP11APR4
                I3=0
170             I01=1
                MISITF=0
                MISTAK=0
                NDIAG=0
                NFATL=0
175             NFDANR = 0                                              SP09APR4
                NFIO=0
                NFSIM = 0                                               SP09APR4
                NMC=0
                NQUA=0
180             NRES=0
                NPL=0
                NROBA=0
                NS=0
                NTL=0
185             NTNR1=0
                NTR=0
                NTRI=0
                NTR1=0
                NTR2 = 0                                                SP08APR4
190             NTR3 = 0                                                SP08APR4
                NUNDI=0
                N2=0
                SEQMAX=0
          C
195             READ 9802,NBOUC,RMISTH
                READ 9803,NMODU,MODSIM
                READ 9802,NSPA,CONDIT
                READ 9805,NONDED,NIO
                READ 9812,PCOM,PBU
200             READ 9807,IROLLA,RECOV,MININT,RACPU
                READ 9808,IDLYRC,RCDUR,RCCINT,DLYCPU
                READ 9809,IROLLB,MAXRLB,RBCPU
                READ 9810,IDESCR,DURMC,RMC,PSMC
                READ 9802,ISTART,DURRES
205             READ 9812,DELAY,TC
                READ 9813,DIAGN,DETMAX,TH2
                READ 9813,PDET,PDM,PDFTIO
                READ 9814,RTI,MINCY,MACY,DOHMAX
                READ 9812,PROMM,PSUC
210             READ 9815,ISYNC,RATINT
                IF(IDESCR.NE.1) IDESCR=2
                IOSIM=NMODU.GE.4
                IF(IOSIM) CALL FCBIN
                IF(IOSIM) CALL FCBPI
215             NSPB=NSPA
                MOD=MODSIM-NSPA
                IF(NONDED.NE.2)NIO=MODSIM
                IDLE=RTI-MINCY
                DURPB=TC
220             IF(IDLYRC.NE.1) GOTO 801
```

```
                  RECOV=PCDUP
                  MININT=RCCINT
                  RACPU=DLYCPU
               RBCPU=DLYCPU
225               DURRB=PCDUR
                  MAXRLB=1
                  IROLLA=1
                  GOTO 805
            801 CONTINUE
230               IF(IROLLB.EQ.1) GOTO 802
                  MAXRLB=0
                  DURRB=0.0
                  GOTO 805
            802 CONTINUE
235               IF(IROLLA.EQ.1) GOTO 805
                  IROLLA=2
                  RECOV=TC
                  RACPU=RBCPU
                  MININT=RTI*MACY
240         805 CONTINUE
          C IF THE EEM'S ARE DEDICATED,AN EEM FAULT MAY HIT THE COMPUTER. IF PBU
          C IS 0, THEN A FAULT IN THE EEM ALWAYS HIT THE COMPUTER (PCOM+PBUCO=1).
          C IF THE EEM'S ARE NON DEDICATED,THEN HTEY NEVER AFFECT A COMPUTER. PCOM
          C AND PBUCO ARE 0.
245               PBUCO=0
                  IF(NONDED.EQ.1) PBUCO=1.0-PCOM-PBU
                  IF(NSPA.EQ.0) PRINT 9900,(NMCONF(I,MOD),I=1,2)
                  IF(NSPA.NE.0) PRINT 9901,(NMCONF(I,MOD),I=1,2),NSPA
                  PRINT 9922,NMIS1
250               PRINT 9923,RMISTH
                  PRINT 9927
                  IF(IDLYRC.NE.1) GOTO 1001
                     PRINT 9902
                     PRINT 9903,RECOV
255                  PRINT 9904,MININT
                     PRINT 9905,RACPU
                  GOTO 1002
            1001 CONTINUE
                  IF(IROLLA.NE.1) GOTO 1002
260                  PRINT 9906
                     PRINT 9903,RECOV
                     PRINT 9904,MININT
                     PRINT 9905,RACPU
            1002 CONTINUE
265               IF(IDESCR.NE.1) GOTO 1003
                     PRINT 9907
                     PRINT 9903,DURMC
                     PRINT 9904,RMC
                     PRINT 9905,PSMC
270         1003 CONTINUE
                  IF(IROLLB.NE.1) GOTO 1004
                     PRINT 9908
                     PRINT 9909,MAXPLB
                     PRINT 9905,RBCPU
275         1004 CONTINUE
```

```
                  PRINT 9910
                  PRINT 9903,OURRES
               PRINT 9911
               PRINT 9912,PSUC
280            PRINT 9913,PDET,PDM,PDETIO
               PRINT 9914,DETMAX,DIAGN
               PRINT 9915,DELAY
               PRINT 9916,TW2
               IF(NSPA.NE.0) PRINT 9917,CONDIT
285            PRINT 9918,RTI,MINCY,MACY,TC,DOHMAX,PROMM
               IF(ISYNC.EQ.0) PRINT 9919,RATINT
               IF(NONDED.EQ.1) PRINT 9020
               IF(NONDFD.EQ.2) PRINT 9921,NIO
               PRINT 9924
290            PRINT 9925,PCOM,PBU,PBUCO
               PPINT 9926
               RMISTM=3600000.*RMISTH
               NMIS=0
               NMIS0=0
295            TIGEN=NBOUC*RMISTM
         C
         C GENERATE FAULTS  FOR NBOUC MISSIONS
            30 CONTINUE
               CALL FAUGEN(MODSIM,TIGEN,300,NMODU)
300            IF (NMIS+NSPA.NE.0) GO TO 56
         C COMPUTE DORMANT FAILURE RATE OF THE COMPUTER
               RMU=RMUP(1)
               IF (NMODU.GE.3) RMU=RMU+RMUP(3)
            56 CONTINUE
305         33 CONTINUE
         C
         C OVERFLOW TEST
               PTR=1
               IF (IFULL.EQ.0) GO TO 40
310            PRINT 9601
               GO TO 10
         C
         C DETERMINE IN WHICH MISSION THE FAULT OCCURS AND THE END OF THE MISSION
         C DETERMINE IF NEW FAULT TABLE IS NEEDED.
315         40 CONTINUE
               K=INT(TABLE(PTR,1)/RMISTM)
               IF (K.GE.NBOUC) GO TO 41
               ENDMIS=K*RMISTM+RMISTM
               NMIS=NMIS0+K
320            IF (NMIS.GE.NMIS1) GO TO 120
               GO TO 43
            41 CONTINUE
               NMIS0=NBOUC*(NMIS/NBOUC)+NBOUC
               NMIS=NMIS0
325            IF (NMIS.GE.NMIS1) GO TO 120
               GO TO 30
         C
         C INITIALIZE MISSION
            43 CONTINUE
330            IFLTCT=0
```

```
                LAUNMI=-10000.
                NWORK=MOD
                NWOIO=NIO
                NSPB=NSPA
                DO 140 J=1,IDIM
                ACFAU(J,3)=0.
            140 CONTINUE
                DO 285 I=1,5
                LAST(I)=-10000.
                COPLAS(I)=-10000.
                NOON(I)=0
                IF (I.LE.MOD) NOON(I)=1
                ISPARE(I)=0
                IF (I.LE.NSPA) ISPARE(I)=MOD+I
                IF (NONDED.NE.2) GO TO 285
                IOCU(I)=0
                IF (I.LE.NWOIO) IOCU(I)=1
            285 CONTINUE
                IF(IOSIM) CALL FCBINI
      C
      C BEGIN SIMULATION
             65 CONTINUE
                GO TO (241,91,66,66,66) MOD
             66 CONTINUE
      C
      C TRIPLEX (NORMAL OPERATION)                                    SP09APR4
                CALL STATE1(NEXT)
             60 CONTINUE
                IF (IDEBUG.NE.1) GO TO 103
                IF (PTR.GE.50) GO TO 103
                PRINT 962,PTR,NEXT,TIME
                DO 102 I=1,IDIM
                IF (ACFAU(I,3).EQ.0.) GO TO 103
                PRINT 985, (ACFAU(I,J),J=1,6)
            102 CONTINUE
            103 CONTINUE
                GO TO (65,70,80,90,100,45,210,220,230,240,250,260),NEXT
             70 CONTINUE
      C
      C ROLLAHEAD RECOVERY                                            SP09APR4
                IF(IROLLA.EQ.0) LAST(FOCO)=TIME
                TIME=TIME+DELAY
                NRL=NRL+1
                CALL STATE2(NEXT)
                GO TO 60
      C
      C SYSTEM RESTART                                                SP10APR4
             80 CONTINUE
                IF(ISTART.EQ.1) GOTO 81
                   ICATAS=ICATAS+1
                   NEXT=5
                   REASON=6
                   GOTO 60
             81 CONTINUE
                NRES=NRES+1
```

```
                CALL STATE3(NEXT)
                GO TO 60
              C
              C DUPLEX (NORMAL OPERATION)                                    SP09APR4
390            90 CONTINUE
                N2=N2+1
               91 CONTINUE                                                   SP03APR4
                CALL STATE4(NEXT)
                GO TO 60
395           C
              C SYSTEM FAILURE                                               SP09APR4
              100 CONTINUE
                NEXT=6
                IF (IDEBUG.NE.1) GO TO 101
400             PRINT 988,PTR,REASON,TIME
              101 CONTINUE
                NFAIL=NFAIL+1
              C SEE DEFINITION OF REASON
                GO TO (85,160,170,180,190,84),REASON                         SP29APR4
405           C
               85 CONTINUE
                PRINT 991,NMIS
                GO TO 50
              160 CONTINUE
410             NFIO=NFIO+1
                GO TO 50
              170 CONTINUE
                NSYSF(IFLTCT)=NSYSF(IFLTCT)+1
                NFCV(IFLTCT)=NFCV(IFLTCT)+1
415             IOF=IOF+1
                GO TO 50
               180  CONTINUE                                                SP09APR4
                NFDANR = NFDANR + 1                                          SP09APR4
                GO TO 50                                                     SP09APR4
420            190  CONTINUE                                                SP09APR4
                NFSIM = NFSIM + 1                                            SP09APR4
                GO TO 50
               84 CONTINUE
                NFAILI(NWORK)=NFAILI(NWORK)+1
425             NTL=NTL+1
               50 CONTINUE
                IF (TABLE(PTR,1).GE.ENDMIS) GO TO 40
                PTR=PTR+1
                GO TO 50
430           C
               45 CONTINUE
                IF(IFLTCT.GE.1) NFCV(IFLTCT)=NFCV(IFLTCT)+1
                GO TO 40
              C
435           C MEMORY COPY RECOVERY                                         SP09APR4
              210 CONTINUE
              C THE NUMBER OF ROLLAHEADS IS DECREASED SINCE A MEMORY COPY IS ALWAYS
              C PRECEDED BY A DUMMY ROLLAHEAD.
                NRL=NRL-1
440           C IF NO MEMORY COPY IN THIS CONFINGURATION DON T DO IT          SP09APR4
```

```
                IF (IDESCR.EQ.2) COPLAS(FOCO)=TIME
                NMC=NMC+1
                CALL  STATE7(NEXT)
                GO TO 60
445           C
              C ROLLBACK RECOVERY                                              SP09APR4
                220 CONTINUE
                IF(MAXRLB.GE.1) NROBA=NROBA+1
                CALL STATE8(NEXT)
450             IF (NEXT.EQ.4) N2=N2-1
                GO TO 60
              C
              C DIAGNOSTICS AND RECOVERY                                       SP09APR4
                230 CONTINUE
455             NDIAG=NDIAG+1
                CALL STATE9(NEXT)                                              SP09APR4
                GO TO 60                                                       SP09APR4
              C
              C SIMPLEX STATE                                                  SP08APR4
460             240 CONTINUE
                ISIM = ISIM + 1                                                SP08APR4
                241 CONTINUE
                CALL STATE8(NEXT)                                              SP08APR4
                GO TO 60                                                       SP09APR4
465           C
              C ROLLBACK IN SIMPLEX
                250 CONTINUE
                IF(MAXRLB.GE.1) GOTO 251
                   NEXT=5
470                REASON=5
                   GOTO 60
                251 CONTINUE
                CALL STATE8(NEXT)
                IF (NEXT.EQ.10) ISIM=ISIM-1
475             GO TO 60
              C
              C INTRODUCE SPARE
                260 CONTINUE
                NS=NS+1
480             CALL STATEC(NEXT)
                GO TO 60
                120 CONTINUE
                PRINT 9400
                NPERM=IFAU-NTR
485             PRINT 9405,NPERM,NTR,IFAU,NFAIL
                PRINT 9410
                GOTO(1105,1104,1103,1102,1101),NOD
               1101 NPERM=NF(5)-NTRF(5)
                PRINT 9411,NPERM,NTRF(5),NF(5)
490            1102 NPERM=NF(4)-NTRF(4)
                PRINT 9412,NPERM,NTRF(4),NF(4)
               1103 NPERM=NF(3)-NTRF(3)
                PRINT 9413,NPERM,NTRF(3),NF(3)
               1104 NPERM=NF(2)-NTRF(2)
495             PRINT 9414,NPERM,NTRF(2),NF(2)
```

```
      1105 NPERM=NF(1)-NTRF(1)
           PRINT 9415,NPERM,NTRF(1),NF(1)
           PRINT 9420
           NQUINT=0
500        GOTO(1110,1109,1108,1107,1106),MOD
      1106 PRINT 9411,NFAILI(5),MISTKI(3),NQUINT
      1107 PRINT 9412,NFAILI(4),MISTKI(2),NQUA
      1108 PRINT 9413,NFAILI(3),MISTKI(1),NTRI
      1109 NDUPFL=NFAILI(2)+NFDANR
505        PRINT 9414,NDUPFL,ITLKP,N2
      1110 PRINT 9415,NFSIM,NTNR1,ISIM
           PRINT 9425
           IF(IDLYRC.NE.1) GOTO 1120
              NDELAY=NROBA+NPL
510           PRINT 9426,NDELAY
              GOTO 1130
      1120 CONTINUE
           IF(IROLLA.EQ.1) PRINT 9427,NRL
           IF(IROLLB.NE.1) GOTO 1130
515        IF(IROLLA.NE.1) NROBA=NROBA+NPL
           PRINT 9428,NROBA
      1130 CONTINUE
           IF(IDESCR.EQ.1) PRINT 9429,NMC
           IF(ISTART.EQ.1) PRINT 9430,NRES
520        PROP=MISITF/(RMISTM*NMIS1/RTI)
           PRINT 9435,PROP,SFQMAX
           IF(NSPA.NE.0) PRINT 9431,NS
           ZC=2.0
           CALL CONFID(ZC,NFAIL,NMIS1,SFP,SFPERR)
525        CALL CONFID(ZC,MISTAK,NTR3+MISTAK,XLEK,XLEK1)
           CALL CONFID(ZC,ITLKP,NTR2+ITLKP,DUPLK,DUPLK1)
           CALL CONFID(ZC,NTNR1,NTNR1+NTR1,SLEK,SLEK1)
           CALL CONFID(ZC,NQUA+NTRI+N2,NQUA+NTRI+N2+ICATAS,XCOV,XCOV1)
           CALL CONFID(ZC,ISIM,ISIM+NDUPFL,DPCOV,DPCOV1)
530        PRINT 9445,SFP,SFPERR
           PRINT 9450,XLEK,XLEK1,DUPLK,DUPLK1,SLEK,SLEK1
           PRINT 9455,XCOV,XCOV1,DPCOV,DPCOV1
           IF(IOSIM) CALL PIOSTS
           GOTO 10
535     20 CONTINUE
           STOP
     2000 CONTINUE
           CALL FCBPM1
           GOTO 10
540  3000 CONTINUE
           CALL FCBPM2
           GOTO 10
          C                  **********************************                  SP02APR4
          C                  *         F O R M A T S        *                  SP02APR4
545       C                  **********************************                  SP02APR4
        962 FORMAT(6H PTR =,I3,3X,6HNEXT =,I3,3X,6HTIME =,F19.3)
        985 FORMAT(1X,6F19.3)
        986 FORMAT(6H PTR =,I3,10H  REASON =,I2,8H  TIME =,F19.3)
        991 FORMAT(25H TOO MANY FAULTS, MISSION,I6)
550    9400 FORMAT(1H1,30X,42HS I M U L A T I O N    S T A T I S T I C S  )
```

```
      9405 FORMAT(/1X,20HNUMBER OF PERMANENTS,5X,I10/1X,
     1 20HNUMBER OF TRANSIENTS,5X,I10/1X,
     2 22HTOTAL NUMBER OF FAULTS,3X,I10/1X,
     3 25HNUMBER OF SYSTEM FAILURES,I10)
555   9410 FORMAT(//1X,33HGPC FAULT AND RECOVERY STATISTICS
     1 //3X,13HCONFIGURATION,5X,16HPERMANENT FAULTS,4X,
     2 16HTRANSIENT FAULTS,4X,12HTOTAL FAULTS)
      9411 FORMAT(5X,10HQUINTUPLEX,6X,I10,10X,I10,10X,I10)
      9412 FORMAT(5X,10HQUADRUPLEX,6X,I10,10X,I10,10X,I10)
560   9413 FORMAT(5X,10HTRIPLEX    ,6X,I10,10X,I10,10X,I10)
      9414 FORMAT(5X,10HDUPLEX     ,6X,I10,10X,I10,10X,I10)
      9415 FORMAT(5X,10HSIMPLEX    ,6X,I10,10X,I10,10X,I10)
      9420 FORMAT(/21X,15HSYSTEM FAILURES,5X,20HLEAKY TRANSIENTS
     1 ,15HDEGRADATIONS TO)
565   9425 FORMAT(/34H GPC RECOVERY PROCEDURE STATISTICS  )
      9426 FORMAT(3X,25HNUMBER OF DELAY-RECOVERYS,3X,I10)
      9427 FORMAT(3X,20HNUMBER OF ROLLAHEADS,8X,I10)
      9428 FORMAT(3X,19HNUMBER OF ROLLBACKS,9X,I10)
      9429 FORMAT(3X,22HNUMBER OF MEMORY COPYS,6X,I10)
570   9430 FORMAT(3X,25HNUMBER OF SYSTEM RESTARTS,3X,I10)
      9431 FORMAT(3X,21HNUMBER OF SPARES USED,7X,I10)
      9435 FORMAT(/3X,31HPROPORTION OF MISSED ITERATIONS,7X,E10.3
     1 /3X,35HLONGEST SERIES OF MISSED ITERATIONS,3X,I10)
      9445 FORMAT(/28H MISSION FAILURE PROBABILITY,3X,F10.8,5H +/- ,F10.8)
575   9450 FORMAT(/9H LEAKAGES/5X,9HMULTIPLEX,2X,F10.8,5H +/- ,F10.8/
     1 5X,6HDUPLEX,5X,F10.8,5H +/- ,F10.8/
     2 5X,7HSIMPLEX,4X,F10.8,5H +/- ,F10.8)
      9455 FORMAT(/10H COVERAGES/5X,9HMULTIPLEX,2X,F10.8,5H +/- ,F10.8/
     1 5X,6HDUPLEX,5X,F10.8,5H +/- ,F10.8)
580   9601 FORMAT(41H SIMULATION ABORTED  FAULT TABLE OVERFLOW)
      9801 FORMAT(2I10)
      9802 FORMAT(I10,F10.0)
      9803 FORMAT(2I10)
      9804 FORMAT(I10,2F10.0)
585   9805 FORMAT(2I10)
      9806 FORMAT(3F10.0)
      9807 FORMAT(I10,3F10.0)
      9808 FORMAT(I10,3F10.0)
      9809 FORMAT(2I10,F10.0)
590   9810 FORMAT(I10,3F10.0)
      9812 FORMAT(2F10.0)
      9813 FORMAT(3F10.0)
      9814 FORMAT(2F10.0,I10,2F10.0)
      9815 FORMAT(I10,F10.0)
595   9900 FORMAT(1H1,30X,2A5)
      9901 FORMAT(1H1,30X,2A5,2H -,I2,7H SPARES)
      9902 FORMAT(/5X,21HDELAY RECONFIGURATION)
      9903 FORMAT(8X,8HDURATION,15X,F10.2,13H MILLISECONDS)
      9904 FORMAT(8X,19HRECURRENCE INTERVAL,4X,F10.2,13H MILLISECONDS)
600   9905 FORMAT(8X,13HEFFECTIVENESS,10X,F10.6)
      9906 FORMAT(/5X,9HROLLAHEAD)
      9907 FORMAT(/5X,11HMEMORY COPY)
      9908 FORMAT(/5X,8HROLLBACK)
      9909 FORMAT(8X,16HNUMBER OF RETRYS,7X,I2)
605   9910 FORMAT(/5X,14HSYSTEM RESTART)
```

```
      9911 FORMAT(//1X,20HRECOVERY  PARAMETERS)
      9912 FORMAT(/5X,21HPROGRAM SURVIVABILITY,5X,F10.6)
      9913 FORMAT(/5X,38HPROBABILITY OF FAULT DETECTION BY BITE
     1 /8X,17HCENTRAL PROCESSOR,6X,F6.3
610   2 /8X,6HMEMORY,17X,F6.3
     3 /8X,13HI/O PROCESSOR,10X,F6.3)
      9914 FORMAT(/5X,14HSTP EFFICIENCY,12X,F10.6
     1 /5X,19HMEAN DIAGNOSIS TIME,7X,F10.2,13H MILLISECONDS)
      9915 FORMAT(5X,21HDELAY BEFORE RECOVERY,5X,F10.2,13H MILLISECONDS)
615   9916 FORMAT(5X,18HISOLATION DURATION,8X,F10.2,13H MILLISECONDS)
      9917 FORMAT(/5X,23HSPARE CONDITIONING TIME,3X,F10.2,13H MILLISECONDS)
      9918 FORMAT(/1X,20HSOFTWARE PAPAMETERS
     1 /5X,26HITERATION PERIOD           ,F10.2,13H MILLISECONDS
     2 /5X,26HMINOR CYCLE DURATION       ,F10.2,13H MILLISECONDS
620  3 /5X,26HMAJOR CYCLE DURATION       ,I10,11H ITERATIONS
     4 /5X,26HTIME BETWEEN COMPARISONS   ,F10.2,13H MILLISECONDS
     5 /5X,26HMAXIMUM DOWN TIME          ,F10.2,13H MILLISECONDS
     6 /5X,26HMINOR CYCLE PROGRAM SIZE   ,F10.6)
      9919 FORMAT(5X,39HASYNCHRONOUS EXECUTIVE - INTERRUPT RATE,F8.1,
625  1 11H PER SECOND)
      9920 FORMAT(/1X,24HDEDICATED I/O PROCESSORS)
      9921 FORMAT(/1X,I2,1X,28HNON-DEDICATED I/O PROCESSORS)
      9922 FORMAT(/1X,18HNUMBER OF MISSIONS,I12)
      9923 FORMAT(1X,16HMISSION DURATION,F14.4,1X,5HHOURS)
630   9924  FORMAT(10H1NOTATIONS
     1 /5X,34HMODULE 1 - CENTRAL PROCESSING UNIT
     2 /5X,24HMODULE 2 - I/O PROCESSOR
     3 /5X,17HMODULE 3 - MEMORY
     4 /5X,27HMODULE 4 - EXTERNAL DEVICES)
635   9925 FORMAT(/1X,30HIMPACT OF I/O PROCESSOR FAULTS
     1 /5X,11HON COMPUTER,10X,F10.6
     2 /5X,6HON BUS,15X,F10.6
     3 /5X,19HON BUS AND COMPUTER,2X,F10.6)
      9926 FORMAT(//1X,36HDESCRIPTION OF THE FAULT ENVIORNMENT)
640   9927 FORMAT(//1X,29HTRANSIENT RECOVERY PROCEDURES)
       END
```

SYMBOLIC REFERENCE MAP

ENTRY POINTS DEF LINE REFERENCES
4050 DRIVER 1

VARIABLES SN TYPE RELOCATION

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 90 | 363 | 364 | DEFINED | 336 | | |
| 24 | AVDUR | REAL | ARRAY | C014 | REFS | 98 | | | | | | |
| 0 | CONDIT | REAL | | C035 | REFS | 109 | 284 | DEFINED | 197 | | | |
| 0 | COPLAS | REAL | ARRAY | COM9 | REFS | 93 | DEFINED | 340 | 441 | | | |
| 2265 | DELAY | REAL | | COM1 | REFS | 88 | 282 | 372 | DEFINED | 205 | | |
| 1 | DEOR | REAL | | C027 | REFS | 103 | DEFINED | 160 | | | | |
| 1 | DETMAX | REAL | | DETE | REFS | 123 | 281 | DEFINED | 206 | | | |
| 0 | DIAGN | REAL | | D | REFS | 121 | 281 | DEFINED | 206 | | | |
| 7031 | DLYCPU | REAL | | | REFS | 223 | 224 | DEFINED | 201 | | | |
| 0 | DOR | REAL | | C027 | REFS | 103 | DEFINED | 159 | | | | |
| 6 | DOHMAX | REAL | | EXEC | REFS | 124 | 285 | DEFINED | 208 | | | |
| 7064 | DPCOV | REAL | | | REFS | 529 | 532 | | | | | |
| 7065 | DPCOV1 | REAL | | | REFS | 529 | 532 | | | | | |
| 7056 | DUPLK | REAL | | | REFS | 526 | 531 | | | | | |
| 7057 | DUPLK1 | REAL | | | REFS | 526 | 531 | | | | | |
| 6 | DURMC | REAL | | COM9 | REFS | 93 | 267 | DEFINED | 203 | | | |
| 1 | DURRB | REAL | | C010 | REFS | 94 | DEFINED | 219 | 225 | 232 | | |
| 0 | DURRES | REAL | | C013 | REFS | 97 | 277 | DEFINED | 204 | | | |
| 74 | ENDMIS | REAL | | COM3 | REFS | 90 | 427 | DEFINED | 318 | | | |
| 2262 | EXTEN | INTEGER | | COM1 | REFS | 88 | 134 | | | | | |
| 0 | FOCO | REAL | | COM8 | REFS | 92 | 371 | 441 | DEFINED | 161 | | |
| 0 | FOCO2 | REAL | | FAILD2 | REFS | 125 | DEFINED | 162 | | | | |
| 5 | FRTI | REAL | | C027 | REFS | 103 | | | | | | |
| 1 | FSHI | REAL | | C040 | REFS | 114 | | | | | | |
| 7036 | I | INTEGER | | | REFS | 247 | 248 | 339 | 340 | 341 | 2*342 | 343 |
| | | | | | 3*344 | 346 | 2*347 | 363 | 364 | DEFINED | 247 | 248 |
| | | | | | 338 | 362 | | | | | | |
| 0 | ICATAS | INTEGER | | C041 | REFS | 115 | 380 | 528 | DEFINED | 163 | 380 | |
| 0 | IDEBUG | INTEGER | | DEBUG | REFS | 122 | 359 | 399 | DEFINED | 151 | | |
| 7033 | IDESCR | INTEGER | | | REFS | 211 | 265 | 441 | 518 | DEFINED | 203 | 211 |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 88 | | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 88 | 335 | 362 | DEFINED | 144 | | |
| 4 | IDLE | REAL | | EXEC | REFS | 124 | 139 | DEFINED | 218 | | | |
| 7026 | IDLYRC | INTEGER | | | REFS | 220 | 252 | 508 | DEFINED | 201 | | |
| 1 | IFAU | INTEGER | | C012 | REFS | 96 | 484 | 485 | DEFINED | 164 | | |
| 0 | IFLTCT | INTEGER | | FLTMS | REFS | 133 | 2*413 | 2*414 | 3*432 | DEFINED | 330 | |
| 1 | IFULL | INTEGER | | COM2 | REFS | 89 | 309 | DEFINED | 165 | | | |
| 1 | IFWE | INTEGER | | COVER | REFS | 119 | | | | | | |
| 0 | IOCU | INTEGER | ARRAY | C038 | REFS | 112 | DEFINED | 346 | 347 | | | |
| 7004 | IOF | INTEGER | | | REFS | 415 | DEFINED | 166 | 415 | | | |
| 7001 | IOSIM | LOGICAL | | | REFS | 136 | 213 | 214 | 349 | 533 | | |
| | | | | | DEFINED | 212 | | | | | | |
| 7006 | IO1 | * INTEGER | | | DEFINED | 170 | | | | | | |
| 7025 | IROLLA | INTEGER | | | REFS | 235 | 259 | 371 | 513 | 515 | | |
| | | | | | DEFINED | 200 | 227 | 236 | | | | |
| 7032 | IROLLB | INTEGER | | | REFS | 230 | 271 | 514 | DEFINED | 202 | | |
| 7005 | ISIM | INTEGER | | | REFS | 461 | 474 | 506 | 2*529 | DEFINED | 167 | 461 |
| | | | | | 474 | | | | | | | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| J | ISPARE | INTEGER | ARRAY | C034 | REFS | 108 | DEFINED | 343 | 344 | | | |
| 7034 | ISTART | INTEGER | | | REFS | 379 | 519 | DEFINED | 204 | | | |
| 0 | ISHI | INTEGER | | C040 | REFS | 114 | | | | | | |
| 1 | ISYNC | INTEGER | | C031 | REFS | 106 | 286 | DEFINED | 210 | | | |
| 0 | ITLKP | INTEGER | | COVER | REFS | 119 | 505 | 2*526 | DEFINED | 168 | | |
| 1 | I3 | INTEGER | | C041 | REFS | 115 | DEFINED | 169 | | | | |
| 7042 | J | INTEGER | | | REFS | 336 | 364 | DEFINED | 335 | 364 | | |
| 7041 | K | INTEGER | | | REFS | 317 | 318 | 319 | DEFINED | 316 | | |
| 0 | LAST | REAL | ARRAY | PERM | REFS | 128 | 139 | DEFINED | 339 | 371 | | |
| 0 | LAUNMI | REAL | | CYC | REFS | 120 | 139 | DEFINED | 331 | | | |
| 1 | MACY | INTEGER | | MM | REFS | 127 | 239 | 285 | DEFINED | 208 | | |
| 0 | MAXRLB | INTEGER | | C010 | REFS | 94 | 273 | 448 | 468 | DEFINED | 202 | 226 |
| | | | | | 231 | | | | | | | |
| 75 | MEMSIZ | INTEGER | | COM3 | REFS | 90 | | | | | | |
| 4 | MI | INTEGER | | C027 | REFS | 103 | | | | | | |
| 0 | MINCY | REAL | | EXEC | REFS | 124 | 139 | 218 | 285 | DEFINED | 208 | |
| 5 | MININT | REAL | | PERM | REFS | 128 | 139 | 255 | 262 | DEFINED | 200 | 222 |
| | | | | | 239 | | | | | | | |
| 2 | MISIT | INTEGER | | C027 | REFS | 103 | | | | | | |
| 7 | MISITE | INTEGER | | EXEC | REFS | 124 | 520 | DEFINED | 171 | | | |
| 0 | MISTAK | INTEGER | | C011 | REFS | 95 | 2*525 | DEFINED | 172 | | | |
| 0 | MISTKI | INTEGER | ARRAY | C042 | REFS | 116 | 137 | 156 | 501 | 502 | 503 | |
| 7035 | MOD | INTEGER | | | REFS | 247 | 248 | 332 | 342 | 344 | 353 | 487 |
| | | | | | 500 | DEFINED | 216 | | | | | |
| 0 | MODSIM | INTEGER | | C028 | REFS | 104 | 216 | 217 | 299 | DEFINED | 196 | |
| 7021 | NBOUC | INTEGER | | | REFS | 295 | 317 | 3*323 | DEFINED | 195 | | |
| 7047 | NDELAY | INTEGER | | | REFS | 510 | DEFINED | 509 | | | | |
| 0 | NDIAG | INTEGER | | C016 | REFS | 100 | 455 | DEFINED | 173 | 455 | | |
| 7046 | NDUPFL | INTEGER | | | REFS | 505 | 529 | DEFINED | 504 | | | |
| 7043 | NEXT | INTEGER | | | REFS | 357 | 361 | 367 | 374 | 386 | 393 | 443 |
| | | | | | 449 | 450 | 456 | 463 | 473 | 474 | 480 | |
| | | | | | DEFINED | 381 | 398 | 469 | | | | |
| 0 | NF | INTEGER | ARRAY | FCOUNT | REFS | 126 | 157 | 488 | 489 | 490 | 491 | 492 |
| | | | | | 493 | 494 | 495 | 496 | 497 | | | |
| 7007 | NFAIL | INTEGER | | | REFS | 402 | 485 | 524 | DEFINED | 174 | 402 | |
| 7066 | NFAILI | INTEGER | ARRAY | | REFS | 137 | 155 | 424 | 501 | 502 | 503 | 504 |
| | | | | | DEFINED | 424 | | | | | | |
| 0 | NFCV | INTEGER | ARRAY | FLTMIS | REFS | 132 | 153 | 414 | 432 | DEFINED | 414 | 432 |
| 7010 | NFDANR | INTEGER | | | REFS | 418 | 504 | DEFINED | 175 | 418 | | |
| 7011 | NFIO | INTEGER | | | REFS | 410 | DEFINED | 176 | 410 | | | |
| 7012 | NFSIM | INTEGER | | | REFS | 421 | 506 | DEFINED | 177 | 421 | | |
| 3 | NIO | INTEGER | | C040 | REFS | 114 | 288 | 333 | DEFINED | 198 | 217 | |
| 0 | NMC | INTEGER | | C029 | REFS | 105 | 442 | 518 | DEFINED | 178 | 442 | |
| 7073 | NMCONF | INTEGER | ARRAY | | REFS | 138 | 247 | 248 | DEFINED | 142 | | |
| 0 | NMIS | INTEGER | | COM2 | REFS | 89 | 300 | 320 | 323 | 325 | 407 | |
| | | | | | DEFINED | 293 | 319 | 324 | | | | |
| 7037 | NMIS0 | INTEGER | | | REFS | 319 | 324 | DEFINED | 294 | 323 | | |
| 7003 | NMIS1 | INTEGER | | | REFS | 152 | 249 | 320 | 325 | 520 | 524 | |
| | | | | | DEFINED | 151 | | | | | | |
| 7023 | NMODU | INTEGER | | | REFS | 212 | 299 | 303 | DEFINED | 196 | | |
| 5 | NONOFO | INTEGER | | C038 | REFS | 112 | 217 | 246 | 287 | 288 | 345 | |
| | | | | | DEFINED | 198 | | | | | | |
| 0 | NOON | INTEGER | ARRAY | C015 | REFS | 99 | DEFINED | 341 | 342 | | | |
| 7044 | NPERM | INTEGER | | | REFS | 485 | 489 | 491 | 493 | 495 | 497 | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | DEFINED | 484 | 488 | 490 | 492 | 494 | 496 | |
| 1 | NOUA | INTEGER | | C017 | REFS | 101 | 502 | 2*528 | DEFINED | 179 | | |
| 7045 | NOUINT | INTEGER | | | REFS | 501 | DEFINED | 499 | | | | |
| 7013 | NRES | INTEGER | | | REFS | 385 | 519 | DEFINED | 180 | 385 | | |
| 7014 | NRL | INTEGER | | | REFS | 373 | 439 | 509 | 513 | 515 | | |
| | | | | | DEFINED | 181 | 373 | 439 | | | | |
| 7015 | NROBA | INTEGER | | | REFS | 448 | 509 | 515 | 516 | DEFINED | 182 | 448 |
| | | | | | 515 | | | | | | | |
| 7016 | NS | INTEGER | | | REFS | 479 | 522 | DEFINED | 183 | 479 | | |
| 7024 | NSPA | INTEGER | | | REFS | 215 | 216 | 247 | 2*248 | 284 | 300 | 334 |
| | | | | | 344 | 522 | DEFINED | 197 | | | | |
| 1 | NSPB | INTEGER | | C028 | REFS | 104 | DEFINED | 215 | 334 | | | |
| 2 | NSWION | INTEGER | | C040 | REFS | 114 | | | | | | |
| 12 | NSYSF | INTEGER | ARRAY | FLTMIS | REFS | 132 | 154 | 413 | DEFINED | 413 | | |
| 7017 | NTL | INTEGER | | | REFS | 425 | DEFINED | 184 | 425 | | | |
| 3 | NTNR1 | INTEGER | | PHILM3 | REFS | 129 | 506 | 2*527 | DEFINED | 185 | | |
| 0 | NTR | INTEGER | | C012 | REFS | 96 | 484 | 485 | DEFINED | 186 | | |
| 5 | NTRF | INTEGER | ARRAY | FCOUNT | REFS | 126 | 158 | 488 | 489 | 490 | 491 | 492 |
| | | | | | 493 | 494 | 495 | 496 | 497 | | | |
| 0 | NTRI | INTEGER | | C017 | REFS | 101 | 503 | 2*528 | DEFINED | 187 | | |
| 2 | NTR1 | INTEGER | | PHILM3 | REFS | 129 | 527 | DEFINED | 188 | | | |
| 1 | NTR2 | INTEGER | | PHILM3 | REFS | 129 | 526 | DEFINED | 189 | | | |
| 0 | NTR3 | INTEGER | | PHILM3 | REFS | 129 | 525 | DEFINED | 190 | | | |
| 1 | NUNDI | INTEGER | | C016 | REFS | 100 | DEFINED | 191 | | | | |
| 6 | NWOIO | INTEGER | | C038 | REFS | 112 | 347 | DEFINED | 333 | | | |
| 5 | NWORK | INTEGER | | C015 | REFS | 99 | 2*424 | DEFINED | 332 | | | |
| 7020 | N2 | INTEGER | | | REFS | 391 | 450 | 505 | 2*528 | DEFINED | 192 | 391 |
| | | | | | 450 | | | | | | | |
| 3 | OLTIS | REAL | | EXEC | REFS | 124 | | | | | | |
| 1 | PBU | REAL | | C046 | REFS | 118 | 246 | 290 | DEFINED | 199 | | |
| 2 | PBUCO | REAL | | C046 | REFS | 118 | 290 | DEFINED | 245 | 246 | | |
| 0 | PCOM | REAL | | C046 | REFS | 118 | 246 | 290 | DEFINED | 199 | | |
| 0 | PDET | REAL | | DETE | REFS | 123 | 280 | DEFINED | 207 | | | |
| 0 | PDETIO | REAL | | C039 | REFS | 113 | 280 | DEFINED | 207 | | | |
| 2 | PDM | REAL | | DETE | REFS | 123 | 280 | DEFINED | 207 | | | |
| 24 | PFRIO | REAL | | FLTMIS | REFS | 132 | | | | | | |
| 0 | PROMM | REAL | | MM | REFS | 127 | 285 | DEFINED | 209 | | | |
| 7050 | PROP | REAL | | | REFS | 521 | DEFINED | 520 | | | | |
| 0 | PSMC | REAL | | C045 | REFS | 117 | 269 | DEFINED | 203 | | | |
| 6 | PSUC | REAL | | PERM | REFS | 128 | 279 | DEFINED | 209 | | | |
| 2261 | PTR | INTEGER | | COM1 | REFS | 88 | 134 | 316 | 360 | 361 | 400 | 427 |
| | | | | | 428 | DEFINED | 308 | 428 | | | | |
| 0 | RACPU | REAL | | RAHEAD | REFS | 130 | 256 | 263 | DEFINED | 200 | 223 | 238 |
| 0 | RATINT | REAL | | C031 | REFS | 106 | 286 | DEFINED | 210 | | | |
| 0 | RBCPU | REAL | | RBACK | REFS | 131 | 238 | 274 | DEFINED | 202 | 224 | |
| 7030 | RCCINT | REAL | | | REFS | 222 | DEFINED | 201 | | | | |
| 7027 | RCOUR | REAL | | | REFS | 221 | 225 | DEFINED | 201 | | | |
| 0 | REASON | INTEGER | | COM7 | REFS | 91 | 134 | 400 | 404 | DEFINED | 382 | 470 |
| 2264 | RECOV | REAL | | COM1 | REFS | 88 | 254 | 261 | DEFINED | 200 | 221 | 237 |
| 0 | RLAMDP | REAL | ARRAY | C014 | REFS | 98 | | | | | | |
| 5 | RLAMDT | REAL | ARRAY | C014 | REFS | 98 | | | | | | |
| 17 | RMAXI | REAL | ARRAY | C014 | REFS | 98 | | | | | | |
| 5 | RMC | REAL | | COM9 | REFS | 93 | 268 | DEFINED | 203 | | | |
| 12 | RMINI | REAL | ARRAY | C014 | REFS | 98 | | | | | | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7022 | RMISTH | REAL | | | REFS | 250 | 292 | DEFINED | 195 | | | |
| 0 | PMISTM | REAL | | C036 | REFS | 110 | 295 | 316 | 2*318 | 520 | | |
| | | | | | DEFINED | 292 | | | | | | |
| 0 | RMU | REAL | | C037 | REFS | 111 | 303 | DEFINED | 302 | 303 | | |
| 0 | RMUP | REAL | ARRAY | C033 | REFS | 107 | 302 | 303 | | | | |
| 1 | RTI | REAL | | EXEC | REFS | 124 | 218 | 239 | 285 | 520 | | |
| | | | | | DEFINED | 208 | | | | | | |
| 7002 | SEED | REAL | | | REFS | 148 | DEFINED | 146 | | | | |
| 5 | SEQMAX | INTEGER | | EXEC | REFS | 124 | 134 | 521 | DEFINED | 193 | | |
| 7052 | SFP | REAL | | | REFS | 524 | 530 | | | | | |
| 7053 | SFPERR | REAL | | | REFS | 524 | 530 | | | | | |
| 7000 | SIMTYP | INTEGER | | | REFS | 135 | 147 | 149 | DEFINED | 146 | | |
| 7060 | SLEK | REAL | | | REFS | 527 | 531 | | | | | |
| 7061 | SLEK1 | REAL | | | REFS | 527 | 531 | | | | | |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 88 | 316 | 427 | | | | |
| 76 | TC | REAL | | COM3 | REFS | 90 | 219 | 237 | 285 | DEFINED | 205 | |
| 25 | TFRIO | REAL | | FLTMIS | REFS | 132 | | | | | | |
| 7040 | TIGEN | REAL | | | REFS | 299 | DEFINED | 295 | | | | |
| 2266 | TIME | REAL | | COM1 | REFS | 88 | 361 | 371 | 372 | 400 | 441 | |
| | | | | | DEFINED | 372 | | | | | | |
| 3 | TISL | REAL | | C027 | REFS | 103 | | | | | | |
| 2 | TODO | REAL | | EXEC | REFS | 124 | | | | | | |
| 0 | TW2 | REAL | | C026 | REFS | 102 | 283 | DEFINED | 206 | | | |
| 7062 | XCOV | REAL | | | REFS | 528 | 532 | | | | | |
| 7063 | XCOV1 | REAL | | | REFS | 528 | 532 | | | | | |
| 7054 | XLEK | REAL | | | REFS | 525 | 531 | | | | | |
| 7055 | XLEK1 | REAL | | | REFS | 525 | 531 | | | | | |
| 7051 | ZC | REAL | | | REFS | 524 | 525 | 526 | 527 | 528 | 529 | |
| | | | | | DEFINED | 523 | | | | | | |

| FILE NAMES | | MODE | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | INPUT | FMT | READS | 146 | 151 | 195 | 196 | 197 | 198 | 199 | 200 |
| | | | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 |
| | | | 210 | | | | | | | | |
| 2022 | OUTPUT | FMT | WRITES | 247 | 248 | 249 | 250 | 251 | 253 | 254 | 255 |
| | | | 256 | 260 | 261 | 262 | 263 | 266 | 267 | 268 | 269 |
| | | | 272 | 273 | 274 | 276 | 277 | 278 | 279 | 280 | 281 |
| | | | 282 | 283 | 284 | 285 | 286 | 287 | 288 | 289 | 290 |
| | | | 291 | 310 | 361 | 364 | 400 | 407 | 483 | 485 | 486 |
| | | | 489 | 491 | 493 | 495 | 497 | 498 | 501 | 502 | 503 |
| | | | 505 | 506 | 507 | 510 | 513 | 516 | 518 | 519 | 521 |
| | | | 522 | 530 | 531 | 532 | | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES | | | | | |
|---|---|---|---|---|---|---|---|---|
| CLFAP | | 2 | 153 | 154 | 155 | 156 | 157 | 158 |
| CONFID | | 5 | 524 | 525 | 526 | 527 | 528 | 529 |
| FAUGEN | | 4 | 299 | | | | | |
| FCBIN | | 0 | 213 | | | | | |
| FCBINI | | 0 | 349 | | | | | |
| FCBPI | | 0 | 214 | | | | | |
| FCBPM1 | | 0 | 538 | | | | | |
| FCBPM2 | | 0 | 541 | | | | | |
| PIOSTS | | 0 | 533 | | | | | |
| RANSET | | 1 | 148 | | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| STATEA | | 1 | 463 |
| STATEB | | 1 | 473 |
| STATEC | | 1 | 480 |
| STATE1 | | 1 | 357 |
| STATE2 | | 1 | 374 |
| STATE3 | | 1 | 386 |
| STATE4 | | 1 | 393 |
| STATE7 | | 1 | 443 |
| STATE8 | | 1 | 449 |
| STATE9 | | 1 | 456 |

| INLINE FUNCTIONS | TYPE | ARGS | DEF LINE | REFERENCES |
|---|---|---|---|---|
| INT | INTEGER | 1 | INTRIN | 316 |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 4052 | 10 | | 145 | 311 | 534 | 539 | 542 | | | | |
| 6030 | 20 | | 535 | 147 | 152 | | | | | | |
| 4775 | 30 | | 298 | 326 | | | | | | | |
| 0 | 33 | INACTIVE | 305 | | | | | | | | |
| 5020 | 40 | | 315 | 309 | 427 | 433 | | | | | |
| 5034 | 41 | | 322 | 317 | | | | | | | |
| 5043 | 43 | | 329 | 321 | | | | | | | |
| 5320 | 45 | | 431 | 367 | | | | | | | |
| 5313 | 50 | | 426 | 408 | 411 | 416 | 419 | 422 | 429 | | |
| 5010 | 56 | | 304 | 300 | | | | | | | |
| 5122 | 60 | | 358 | 375 | 383 | 387 | 394 | 444 | 451 | 457 | 464 | 471 |
| | | | | 475 | 481 | | | | | | |
| 5107 | 65 | | 352 | 367 | | | | | | | |
| 5120 | 66 | | 354 | 3*353 | | | | | | | |
| 5203 | 70 | | 368 | 367 | | | | | | | |
| 5216 | 80 | | 378 | 367 | | | | | | | |
| 5225 | 81 | | 384 | 379 | | | | | | | |
| 5310 | 84 | | 423 | 404 | | | | | | | |
| 5267 | 85 | | 406 | 404 | | | | | | | |
| 5231 | 90 | | 390 | 367 | | | | | | | |
| 5233 | 91 | | 392 | 353 | | | | | | | |
| 5236 | 100 | | 397 | 367 | | | | | | | |
| 5253 | 101 | | 401 | 399 | | | | | | | |
| 0 | 102 | | 365 | 362 | | | | | | | |
| 5163 | 103 | | 366 | 359 | 360 | 363 | | | | | |
| 5405 | 120 | | 482 | 320 | 325 | | | | | | |
| 0 | 140 | | 337 | 335 | | | | | | | |
| 5275 | 160 | | 409 | 404 | | | | | | | |
| 5277 | 170 | | 412 | 404 | | | | | | | |
| 5304 | 180 | | 417 | 404 | | | | | | | |
| 5306 | 190 | | 420 | 404 | | | | | | | |
| 5326 | 210 | | 436 | 367 | | | | | | | |
| 5341 | 220 | | 447 | 367 | | | | | | | |
| 5354 | 230 | | 454 | 367 | | | | | | | |
| 5360 | 240 | | 460 | 367 | | | | | | | |
| 5362 | 241 | | 462 | 353 | | | | | | | |
| 5365 | 250 | | 467 | 367 | | | | | | | |
| 5372 | 251 | | 472 | 468 | | | | | | | |
| 5401 | 260 | | 478 | 367 | | | | | | | |

| STATEMENT LABELS | | | | DFF LINE | REFERENCES | | | |
|---|---|---|---|---|---|---|---|---|
| 5102 | 285 | | | 348 | 338 | 345 | | |
| 4414 | 801 | | | 229 | 220 | | | |
| 4421 | 802 | | | 234 | 230 | | | |
| 4432 | 805 | | | 240 | 228 | 233 | 235 | |
| 6142 | 962 | FMT | | 546 | 361 | | | |
| 6150 | 985 | FMT | | 547 | 364 | | | |
| 6152 | 988 | FMT | | 548 | 400 | | | |
| 6160 | 991 | FMT | | 549 | 407 | | | |
| 4074 | 1000 | | | 150 | 149 | | | |
| 4530 | 1001 | | | 258 | 252 | | | |
| 4555 | 1002 | | | 264 | 257 | 259 | | |
| 4602 | 1003 | | | 270 | 265 | | | |
| 4622 | 1004 | | | 275 | 271 | | | |
| 5441 | 1101 | | | 488 | 487 | | | |
| 5454 | 1102 | | | 490 | 487 | | | |
| 5467 | 1103 | | | 492 | 487 | | | |
| 5502 | 1104 | | | 494 | 487 | | | |
| 5515 | 1105 | | | 496 | 487 | | | |
| 5545 | 1106 | | | 501 | 500 | | | |
| 5556 | 1107 | | | 502 | 500 | | | |
| 5567 | 1108 | | | 503 | 500 | | | |
| 5600 | 1109 | | | 504 | 500 | | | |
| 5613 | 1110 | | | 506 | 500 | | | |
| 5642 | 1120 | | | 512 | 508 | | | |
| 5666 | 1130 | | | 517 | 511 | 514 | | |
| 6032 | 2000 | | | 537 | 149 | | | |
| 6035 | 3000 | | | 540 | 149 | | | |
| 6165 | 9400 | FMT | | 550 | 483 | | | |
| 6174 | 9405 | FMT | | 551 | 485 | | | |
| 6213 | 9410 | FMT | | 555 | 486 | | | |
| 6231 | 9411 | FMT | | 558 | 489 | 501 | | |
| 6236 | 9412 | FMT | | 559 | 491 | 502 | | |
| 6243 | 9413 | FMT | | 560 | 493 | 503 | | |
| 6250 | 9414 | FMT | | 561 | 495 | 505 | | |
| 6255 | 9415 | FMT | | 562 | 497 | 506 | | |
| 6262 | 9420 | FMT | | 563 | 498 | | | |
| 6272 | 9425 | FMT | | 565 | 507 | | | |
| 6277 | 9426 | FMT | | 566 | 510 | | | |
| 6304 | 9427 | FMT | | 567 | 513 | | | |
| 6311 | 9428 | FMT | | 568 | 516 | | | |
| 6316 | 9429 | FMT | | 569 | 518 | | | |
| 6323 | 9430 | FMT | | 570 | 519 | | | |
| 6330 | 9431 | FMT | | 571 | 522 | | | |
| 6335 | 9435 | FMT | | 572 | 521 | | | |
| 6350 | 9445 | FMT | | 574 | 530 | | | |
| 6357 | 9450 | FMT | | 575 | 531 | | | |
| 6374 | 9455 | FMT | | 578 | 532 | | | |
| 6406 | 9601 | FMT | | 580 | 310 | | | |
| 6414 | 9801 | FMT | | 581 | 151 | | | |
| 6416 | 9802 | FMT | | 582 | 146 | 195 | 197 | 204 |
| 6421 | 9803 | FMT | | 583 | 196 | | | |
| 6423 | 9804 | FMT | NO REFS | 584 | | | | |
| 6426 | 9805 | FMT | | 585 | 198 | | | |
| 6430 | 9806 | FMT | NO REFS | 586 | | | | |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | | | |
|---|---|---|---|---|---|---|---|
| 6432 | 9807 | FMT | 587 | 200 | | | |
| 6435 | 9808 | FMT | 588 | 201 | | | |
| 6440 | 9809 | FMT | 589 | 202 | | | |
| 6443 | 9810 | FMT | 590 | 203 | | | |
| 6446 | 9812 | FMT | 591 | 199 | 205 | 209 | |
| 6450 | 9813 | FMT | 592 | 206 | 207 | | |
| 6452 | 9814 | FMT | 593 | 208 | | | |
| 6455 | 9815 | FMT | 594 | 210 | | | |
| 6460 | 9900 | FMT | 595 | 247 | | | |
| 6463 | 9901 | FMT | 596 | 248 | | | |
| 6467 | 9902 | FMT | 597 | 253 | | | |
| 6473 | 9903 | FMT | 598 | 254 | 261 | 267 | 277 |
| 6500 | 9904 | FMT | 599 | 255 | 262 | 268 | |
| 6506 | 9905 | FMT | 600 | 256 | 263 | 269 | 274 |
| 6512 | 9906 | FMT | 601 | 260 | | | |
| 6515 | 9907 | FMT | 602 | 266 | | | |
| 6520 | 9908 | FMT | 603 | 272 | | | |
| 6523 | 9909 | FMT | 604 | 273 | | | |
| 6527 | 9910 | FMT | 605 | 276 | | | |
| 6533 | 9911 | FMT | 606 | 278 | | | |
| 6537 | 9912 | FMT | 607 | 279 | | | |
| 6544 | 9913 | FMT | 608 | 280 | | | |
| 6561 | 9914 | FMT | 612 | 281 | | | |
| 6572 | 9915 | FMT | 614 | 282 | | | |
| 6601 | 9916 | FMT | 615 | 283 | | | |
| 6607 | 9917 | FMT | 616 | 284 | | | |
| 6616 | 9918 | FMT | 617 | 285 | | | |
| 6661 | 9919 | FMT | 624 | 286 | | | |
| 6671 | 9920 | FMT | 626 | 287 | | | |
| 6676 | 9921 | FMT | 627 | 288 | | | |
| 6704 | 9922 | FMT | 628 | 249 | | | |
| 6710 | 9923 | FMT | 629 | 250 | | | |
| 6715 | 9924 | FMT | 630 | 289 | | | |
| 6735 | 9925 | FMT | 635 | 290 | | | |
| 6752 | 9926 | FMT | 639 | 291 | | | |
| 6760 | 9927 | FMT | 640 | 251 | | | |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|
| 5054 | 140 | J | 335 337 | 28 | INSTACK |
| 5063 | 285 | * I | 338 348 | 220 | OPT |
| 5142 | 102 | * I | 362 365 | 218 | EXT REFS EXITS NOT INNER |
| 5150 | | * J | 364 | 78 | EXT REFS |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 IOIM (1) | 1 TABLE (1200) | 1201 PTR (1) | | | |
| | | 1202 EXTEN (1) | 1203 IDETEC (1) | 1204 RECOV (1) | | | |
| | | 1205 DELAY (1) | 1206 TIME (1) | | | | |
| COM2 | 2 | 0 NMIS (1) | 1 IFULL (1) | | | | |
| COM3 | 63 | 0 ACFAU (60) | 60 ENDMIS (1) | 61 MEMSIZ (1) | | | |
| | | 62 TC (1) | | | | | |
| COM7 | 1 | 0 REASON (1) | | | | | |
| COM8 | 1 | 0 FOCO (1) | | | | | |
| COM9 | 7 | 0 COPLAS (5) | 5 RMC (1) | 6 DURMC (1) | | | |
| CO10 | 2 | 0 MAXRLB (1) | 1 DURRB (1) | | | | |

```
COMMON BLOCKS   LENGTH     MEMBERS - 3IAS NAME(LENGTH)
      C011         1              0 MISTAK (1)
      C012         2              0 NTR    (1)              1 IFAU   (1)
      C013         1              0 DURREF (1)
      C014        25              0 RLAMDP (5)              5 RLAMDT (5)             10 RMINI  (5)
                                 15 RMAXI  (5)             20 AVDUR  (5)
      C015         6              0 NOON   (5)              5 NWORK  (1)
      C016         2              0 NDIAG  (1)              1 NUNDI  (1)
      C017         2              0 NTPI   (1)              1 NQUA   (1)
      C026         1              0 TM2    (1)
      C027         6              0 DOR    (1)              1 DEOR   (1)              2 MISIT  (1)
                                  3 TISL   (1)              4 MI     (1)              5 FRTI   (1)
      C028         2              0 MODSIM (1)              1 NSPB   (1)
      C029         1              0 NMC    (1)
      C031         2              0 RATINT (1)              1 ISYNC  (1)
      C033         5              0 RMUP   (5)
      C034         5              0 ISPARE (5)
      C035         1              0 CONDIT (1)
      C036         1              0 RMISTM (1)
      C037         1              0 RMU    (1)
      C038         7              0 IOCU   (5)              5 NONDED (1)              6 NWOIO  (1)
      C039         1              0 PDETIO (1)
      C040         4              0 ISHI   (1)              1 FSHI   (1)              2 NSWION (1)
                                  3 NIO    (1)
      C041         2              0 TCATAS (1)              1 I3     (1)
      C042         3              0 MISTKI (3)
      C045         1              0 PSMC   (1)
      C046         3              0 PCOM   (1)              1 PBU    (1)              2 PBUCO  (1)
      COVER        2              0 TTLKP  (1)              1 IFWE   (1)
      CYC          1              0 LAUNMI (1)
      D            1              0 DIAGN  (1)
      DEBUG        1              0 IDEBUG (1)
      DFTE         3              0 PDFT   (1)              1 DETMAX (1)              2 PDM    (1)
      EXEC         8              0 MINCY  (1)              1 RTI    (1)              2 TODO   (1)
                                  3 OLTIS  (1)              4 IDLE   (1)              5 SEQMAX (1)
                                  6 OOWMAX (1)              7 MISITE (1)
      FAILD2       1              0 FOCO?  (1)
      FCOUNT      10              0 NF     (5)              5 NTRF   (5)
      MM           2              0 PROMM  (1)              1 MACY   (1)
      PERM         7              0 LAST   (5)              5 MININT (1)              6 PSUC   (1)
      PHILM3       4              0 NTP3   (1)              1 NTR2   (1)              2 NTR1   (1)
                                  3 NTNR1  (1)
      PAHEAD       1              0 PACPU  (1)
      RBACK        1              0 PBCPU  (1)
      FLTMIS      22              0 NFCV   (10)            10 NSYSF  (10)            20 PFRID  (1)
                                 21 TFRIO  (1)
      FLTMS        1              0 IFLTCT (1)

STATISTICS
  PROGRAM LENGTH   30438    1571
   BUFFER LENGTH   40448    2084
   COMMON LENGTH   26318    1433
```

```
      SUBROUTINE ASYNC(TIME,FINPR,NEXT)
C                                   THIS VERSION: 29 AUGUST 1974
C ASYNC GENERATES SEVERAL FAULTS IN ACFAU DEPENDING ON THE NUMBER OF
C INTERRUPTS RECEIVED BETWEEN OCCURRENCE AND DETECTION OF THE FAULT
C IN CASE OF AN ASYNCHRONOUS EXECUTIVE, THIS SUBROUTINE CREATES PSEUDO-
C FAULTS CORRESPONDING TO THE FACT THAT A FAULT MAY BE DETECTED MORE
C THAN ONCE
C FINPR   LAST DETECTION (COMPUTED BY DETTIME)
C TIME    OCCURRENCE TIME OF THE FAULT
C NEXT    SET TO 5 IF MORE FAULTS ARE GENERATED THAN CAN BE ACCOMODATED
C         BY ACFAU  (IF IT HAPPENS, INCREASE IDIM AND SIZE OF ACFAU.
C
C ***************************************************************
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,T
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/COM7/REASON
      COMMON/CO31/RATINT,ISYNC
      COMMON/EXEC/MINCY,RTI,TODO,OLTIS,IDLE,SEQMAX,DOHMAX,MISITE
      INTEGER EXTEN,PTR,REASON
      REAL MINCY,IDLE
C ***************************************************************
C
      IF (RANF(0.).LT.MINCY/RTI) RETURN
      IF (RATINT.EQ.0.)RETURN
C TEST IF AN INTERRUPT COMES BEFORE FINPR
      U=0.
   10 CONTINUE
      Y=RANF(0.)
      IF (Y.EQ.0.) GO TO 10
      U=U+(-1000./RATINT)*ALOG(Y)
      IF (U.GT.FINPR-TIME) RETURN
C
C AN INTERRUPT COMES CREATE A NEW FAULT
      DO 20 I=1,IDIM
      IF (ACFAU(I,3).EQ.0.) GO TO 30
   20 CONTINUE
C THE ACFAU TABLE IS FULL
      NEXT=5
      REASON=1
      RETURN
C COPY NEWLY CREATED FAULT
   30 CONTINUE
      ACFAU(I,2)=TABLE(PTR,1)+TABLE(PTR,2)
      ACFAU(I,3)=1.
      ACFAU(I,4)=TABLE(PTR,4)
      ACFAU(I,5)=0.
      CALL DETTIM(DETEC,TIME+U,1.0)
      ACFAU(I,6)=DETEC
      IF (DETEC.LT.FINPR)FINPR=DETEC
      GO TO 10
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | ASYNC | 1 | 23 | 24 | 31 | 40 | | | | | | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 15 | 35 | DEFINED | 43 | 44 | 45 | 46 |
| | | | | | 48 | | | | | | | |
| 2265 | DELAY | REAL | | COM1 | REFS | 14 | | | | | | |
| 106 | DETEC | REAL | | | REFS | 47 | 48 | 2*49 | | | | |
| 6 | DOWMAX | REAL | | EXEC | REFS | 18 | | | | | | |
| 74 | ENDMIS | REAL | | COM3 | REFS | 15 | | | | | | |
| 2262 | EXTEN | INTEGER | | COM1 | REFS | 14 | 19 | | | | | |
| 0 | FINPR | REAL | | F.P. | REFS | 31 | 49 | DEFINED | 1 | 49 | | |
| 105 | I | INTEGER | | | REFS | 35 | 43 | 44 | 45 | 46 | 48 | |
| | | | | | DEFINED | 34 | | | | | | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 14 | | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 14 | 34 | | | | | |
| 4 | IDLE | REAL | | EXEC | REFS | 18 | 20 | | | | | |
| 1 | ISYNC | INTEGER | | COM1 | REFS | 17 | | | | | | |
| 75 | MEMSIZ | INTEGER | | COM3 | REFS | 15 | | | | | | |
| 0 | MINCY | REAL | | EXEC | REFS | 18 | 20 | 23 | | | | |
| 7 | MISITE | INTEGER | | EXEC | REFS | 18 | | | | | | |
| 0 | NEXT | INTEGER | | F.P. | DEFINED | 1 | 38 | | | | | |
| 3 | OLTIS | REAL | | EXEC | REFS | 18 | | | | | | |
| 2261 | PTR | INTEGER | | COM1 | REFS | 14 | 19 | 2*43 | 45 | | | |
| 0 | RATINT | REAL | | COM1 | REFS | 17 | 24 | 30 | | | | |
| 0 | REASON | INTEGER | | COM7 | REFS | 16 | 19 | DEFINED | 39 | | | |
| 2264 | RECOV | REAL | | COM1 | REFS | 14 | | | | | | |
| 1 | RTI | REAL | | EXEC | REFS | 18 | 23 | | | | | |
| 5 | SEQMAX | REAL | | EXEC | REFS | 18 | | | | | | |
| 2266 | T | REAL | | COM1 | REFS | 14 | | | | | | |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 14 | 2*43 | 45 | | | | |
| 76 | TC | REAL | | COM3 | REFS | 15 | | | | | | |
| 0 | TIME | REAL | | F.P. | REFS | 31 | 47 | DEFINED | 1 | | | |
| 2 | TODO | REAL | | EXEC | REFS | 18 | | | | | | |
| 103 | U | REAL | | | REFS | 30 | 31 | 47 | DEFINED | 26 | 30 | |
| 104 | Y | REAL | | | REFS | 29 | 30 | DEFINED | 28 | | | |

| EXTERNALS | TYPE | ARGS | | REFERENCES | |
|---|---|---|---|---|---|
| ALOG | REAL | 1 | LIBRARY | 30 | |
| DETTIM | | 3 | | 47 | |
| RANF | REAL | 1 | | 23 | 28 |

| STATEMENT LABELS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|
| 17 | 10 | 27 | 29 | 50 |
| 0 | 20 | 36 | 34 | |
| 51 | 30 | 42 | 35 | |

| LOOPS | LABEL | | INDEX | FROM-TO | LENGTH | PROPERTIES | |
|---|---|---|---|---|---|---|---|
| 41 | 20 | * | I | 34 36 | 48 | INSTACK | EXITS |

| COMMON BLOCKS | LENGTH. | MEMBERS - BIAS | NAME | (LENGTH) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 | IDIM | (1) | 1 | TABLE | (1200) | 1201 | PTR | (1) | |
| | | 1202 | EXTEN | (1) | 1203 | IDETEC | (1) | 1204 | RECOV | (1) | |
| | | 1205 | DELAY | (1) | 1206 | T | (1) | | | | |
| COM3 | 63 | 0 | ACFAU | (60) | 60 | ENDMIS | (1) | 61 | MEMSIZ | (1) | |
| | | 62 | TC | (1) | | | | | | | |
| COM7 | 1 | 0 | REASON | (1) | | | | | | | |
| CO31 | 2 | 0 | RATINT | (1) | 1 | ISYNC | (1) | | | | |
| EXEC | 8 | 0 | MINCY | (1) | 1 | RTI | (1) | 2 | TODO | (1) | |
| | | 3 | OLTIS | (1) | 4 | IDLE | (1) | 5 | SEQMAX | (1) | |
| | | 6 | DOWMAX | (1) | 7 | MISITE | (1) | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 107B | 71 |
| COMMON LENGTH | 24018 | 1281 |

```
      SUBROUTINE BUSCHK(NEXT,NUNIT)
C                        THIS VERSION:   MARCH 1976
      COMMON/FAULT/FLTTYP
      COMMON/COM7/REASON
 5    COMMON/FCBCNT/FCBSF,FALFCB(50)
      COMMON/STATUS/STS(20)
      INTEGER FLTTYP,REASON,FCBSF,FALFCB,STS
     *IBUS=IRAN(1,8)
      CALL BUSFLT(IBUS)
10    IF(STS(1).EQ.0) RETURN
      REASON=7
      NEXT=5
      FCBSF=FCBSF+1
      K=STS(2)
15    FALFCB(K)=FALFCB(K)+1
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|
| 2 | BUSCHK | 1 | 10 | 16 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | FALFCB | INTEGER | ARRAY | FCBCNT | REFS | 5 | 7 | 15 | DEFINED | 15 | |
| 0 | FCBSF | INTEGER | | FCBCNT | REFS | 5 | 7 | 13 | DEFINED | 13 | |
| 0 | FLTTYP | INTEGER | | FAULT | REFS | 3 | 7 | | | | |
| 32 | IBUS | INTEGER | | | REFS | 9 | DEFINED | 8 | | | |
| 33 | K | INTEGER | | | REFS | 2*15 | DEFINED | 14 | | | |
| 0 | NEXT | INTEGER | | F.P. | DEFINED | 1 | 12 | | | | |
| 0 | NUNIT | INTEGER | *UNUSED | F.P. | DEFINED | 1 | | | | | |
| 0 | REASON | INTEGER | | COM7 | REFS | 4 | 7 | DEFINED | 11 | | |
| 0 | STS | INTEGER | ARRAY | STATUS | REFS | 6 | 7 | 10 | 14 | | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| BUSFLT | | 1 | 9 |
| IRAN | INTEGER | 2 | 8 |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | |
|---|---|---|---|
| FAULT | 1 | 0 FLTTYP (1) | |
| COM7 | 1 | 0 REASON (1) | |
| FCBCNT | 51 | 0 FCBSF (1) | 1 FALFCB (50) |
| STATUS | 20 | 0 STS (20) | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 34B | 28 |
| COMMON LENGTH | 111B | 73 |

```
      SUBROUTINE BUSFLT(BUS)
C
C THIS SUBROUTINE DETERMINES THE EFFECT OF A BUS FAULT ON THE FLIGHT
C CRITICAL BUS EQUIPMENT GROUP.
C
C BUS    IDENTIFIES THE FAULTY BUS
C
C*************************************************************
      COMMON/FCB1/BUSSTS(8),FCB(8,10),BTUSTS(10)
      COMMON/FCB2/NBTU,BTUTYP(10),BTUCON(10,4),BTUNO(10)
      COMMON/FCB3/BUSTFL,BUSCOV
      COMMON/FCB4/NBUSTF,NBUSPF,NBUSTR
      COMMON/CO12/NTR,IFAU
      COMMON/FAULT/FLTTYP
      COMMON/STATUS/STS(20)
      COMMON/FCBUC/FCBUCF(6)
      COMMON/FLAGS/FLG(5)
      LOGICAL FLG
      LOGICAL GREATR
      INTEGER FCBUCF
      INTEGER BTUNO,BUSSTS,FCB,BTUSTS
      INTEGER  NBTU,BTUTYP,BTUCON,FLTTYP,STS
      INTEGER  BUS,BSTS
C
C                ...IGNORE THE FAULT IF THE BUS IS ALREADY DISABLED
      BSTS = BUSSTS(BUS)
      IF(BSTS.EQ.0) RETURN
      FLG(1)=.TRUE.
C                ...JUMP TO TRANSIENT OR PERMANENT FAULT ROUTINE
      GOTO(100,200,205),FLTTYP
C
C TRANSIENT FAULT HAS OCCURED
C
  100 CONTINUE
C                ...INCREMENT FAULT COUNTERS
      IFAU = IFAU+1
      NTR = NTR.+1
      NBUSTF = NBUSTF+1
C                ...IF TRANSIENT RECOVERY FAILS, GOTO PERMANENT RECOV
      IF(GREATR(BUSTFL)) RETURN
      NBUSTR = NBUSTR+1
      GOTO 205
C
C PERMANENT FAULT RECOVERY
C
  200 CONTINUE
      IFAU = IFAU+1
      NBUSPF = NBUSPF+1
C                ...ENTER LEAKY TRANSIENTS
  205 CONTINUE
      DO 300 J=1,NBTU
        ITYP=FCB(BUS,J)+1
        GOTO(300,400,500),ITYP
  300 CONTINUE
      BUSSTS(BUS)=0
```

```
      IF(STS(1).NE.0) CALL SETSTS(24)
      RETURN
C
C                ...AN ACTIVE BTU PORT FAILS
  400 CONTINUE
      MDM=J
      FCB(BUS,MDM)=0
C                ...IF THE BTU IS DISABLED, GOTO 450
      NL = BTUSTS(MDM)-1
      BTUSTS(MDM) = NL
      IF( NL.EQ.0) GOTO  450
C                ...GOTO 300, IF IT HAS A DDU PORT THAT FAILED
      IF(BTUTYP(MDM).EQ.1) GOTO 300
C                ...IF THE BUS SWITCHING CAUSES A SYSTEM FAILURE, 900
      IF(GREATR(BUSCOV)) GOTO 900
C                ...ELSE, SWITCH TO BACKUP PORT
      DO 420 I=1,2
        IBUS=BTUCON(MDM,I)
        IF(FCB(IBUS,MDM).EQ.2)GOTO 425
  420 CONTINUE
C                ...IF CONTROL FALLS THROUGH, SOMETHING IS WRONG.
      PRINT 1000,BTUCON
 1000 FORMAT(/12H ***ERROR***/1X,4(10I5/)/)
      CALL PIOCNF
C
C                ...INDICATE THAT BACKUP PORT IS NOW ACTIVE
  425 FCB(IBUS,MDM)=1
      GOTO 300
C
C                ...AN MDM IS DISABLED
  450 CONTINUE
      CALL MDMPF(MDM)
      GOTO 300
C
C                ...AN INACTIVE PORT HAS FAILED
  500 CONTINUE
      FCB(BUS,J) = 0
      BTUSTS(J)=BTUSTS(J)-1
      GOTO 300
  900 CONTINUE
      CALL SETSTS(FCBUCF(1))
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | BUSFLT | 1 | 27 | 40 | 57 | 97 | | | | | | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 173 | BSTS | INTEGER | | | REFS | 23 | 27 | DEFINED | 26 | | | | |
| 13 | BTUCON | INTEGER | ARRAY | FCB2 | REFS | 10 | 22 | 73 | 77 | | | | |
| 63 | BTUNO | INTEGER | ARRAY | FCB2 | REFS | 10 | 21 | | | | | | |
| 130 | BTUSTS | INTEGER | ARRAY | FCB1 | REFS | 9 | 21 | 64 | 93 | DEFINED | 65 | 93 | |
| 1 | BTUTYP | INTEGER | ARRAY | FCB2 | REFS | 10 | 22 | 68 | | | | | |
| 0 | BUS | INTEGER | | F.P. | REFS | 23 | 26 | 52 | 55 | 62 | 92 | | |
| | | | | | DEFINED | 1 | | | | | | | |
| 1 | BUSCOV | REAL | | FCB3 | REFS | 11 | 70 | | | | | | |
| 0 | BUSSTS | INTEGER | ARRAY | FCB1 | REFS | 9 | 21 | 26 | DEFINED | 55 | | | |
| 0 | BUSTFL | REAL | | FCB3 | REFS | 11 | 40 | | | | | | |
| 10 | FCB | INTEGER | ARRAY | FCB1 | REFS | 9 | 21 | 52 | 74 | DEFINED | 62 | 82 | |
| | | | | | 92 | | | | | | | | |
| 0 | FCBUCF | INTEGER | ARRAY | FCBUC | REFS | 16 | 20 | 96 | | | | | |
| 0 | FLG | LOGICAL | ARRAY | FLAGS | REFS | 17 | 18 | DEFINED | 28 | | | | |
| 0 | FLTTYP | INTEGER | | FAULT | REFS | 14 | 22 | 30 | | | | | |
| 200 | I | INTEGER | | | REFS | 73 | DEFINED | 72 | | | | | |
| 201 | IBUS | INTEGER | | | REFS | 74 | 82 | DEFINED | 73 | | | | |
| 1 | IFAU | INTEGER | | CO12 | REFS | 13 | 36 | 47 | DEFINED | 36 | 47 | | |
| 175 | ITYP | INTEGER | | | REFS | 53 | DEFINED | 52 | | | | | |
| 174 | J | INTEGER | | | REFS | 52 | 61 | 92 | 2*93 | DEFINED | 51 | | |
| 176 | MDM | INTEGER | | | REFS | 62 | 64 | 65 | 68 | 73 | 74 | 82 | |
| | | | | | 87 | DEFINED | 61 | | | | | | |
| 0 | NBTU | INTEGER | | FCB2 | REFS | 10 | 22 | 51 | | | | | |
| 1 | NBUSPF | INTEGER | | FCB4 | REFS | 12 | 48 | DEFINED | 48 | | | | |
| 0 | NBUSTF | INTEGER | | FCB4 | REFS | 12 | 38 | DEFINED | 38 | | | | |
| 2 | NBUSTR | INTEGER | | FCB4 | REFS | 12 | 41 | DEFINED | 41 | | | | |
| 177 | NL | INTEGER | | | REFS | 65 | 66 | DEFINED | 64 | | | | |
| 0 | NTR | INTEGER | | CO12 | REFS | 13 | 37 | DEFINED | 37 | | | | |
| 0 | STS | INTEGER | ARRAY | STATUS | REFS | 15 | 22 | 56 | | | | | |

| FILE NAMES | MODE | | | |
|---|---|---|---|---|
| OUTPUT | FMT | WRITFS | 77 | |

| EXTERNALS | TYPE | ARGS | REFERENCES | | | |
|---|---|---|---|---|---|---|
| GREATR | LOGICAL | 1 | 19 | 40 | 70 | |
| MDMPF | | 1 | 87 | | | |
| PIOCNF | | 0 | 79 | | | |
| SETSTS | | 1 | 56 | 96 | | |

| STATEMENT LABELS | | DEF LINE | REFERENCES | | | | | |
|---|---|---|---|---|---|---|---|---|
| 33 | 100 | 34 | 30 | | | | | |
| 44 | 200 | 46 | 30 | | | | | |
| 47 | 205 | 50 | 30 | 42 | | | | |
| 62 | 300 | 54 | 51 | 53 | 68 | 83 | 88 | 94 |
| 74 | 400 | 60 | 53 | | | | | |
| 0 | 420 | 75 | 72 | | | | | |
| 133 | 425 | 82 | 74 | | | | | |
| 136 | 450 | 86 | 66 | | | | | |

| STATEMENT LABELS | | | DEF LINE | REFERENCES |
|---|---|---|---|---|
| 141 | 500 | | 91 | 53 |
| 146 | 900 | | 95 | 70 |
| 164 | 1000 | FMT | 78 | 77 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | ENTRIES | EXITS |
|---|---|---|---|---|---|---|---|
| 50 | 300 | * J | 51 54 | 158 | | | |
| 116 | 420 | * I | 72 75 | 68 | INSTACK | EXITS | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| FCB1 | 98 | 0 BUSSTS (8) | | 8 FCB | (80) | 88 BTUSTS | (10) |
| FCB2 | 61 | 0 NBTU | (1) | 1 BTUTYP | (10) | 11 BTUCON | (40) |
| | | 51 BTUNO | (10) | | | | |
| FCB3 | 2 | 0 BUSTFL | (1) | 1 BUSCOV | (1) | | |
| FCB4 | 3 | 0 NBUSTF | (1) | 1 NBUSPF | (1) | 2 NBUSTR | (1) |
| C012 | 2 | 0 NTR | (1) | 1 IFAU | (1) | | |
| FAULT | 1 | 0 FLTTYP | (1) | | | | |
| STATUS | 20 | 0 STS | (20) | | | | |
| FCBUC | 6 | 0 FCBUCF | (6) | | | | |
| FLAGS | 5 | 0 FLG | (5) | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 2048 | 132 |
| COMMON LENGTH | 3068 | 198 |

```
      SUBROUTINE CONFID(ZC,NS,N,PM,PI)
      PM=-1.0
      PI=-1.0
      IF(N.EQ.0) RETURN
5     XN=FLOAT(N)
      P=NS/XN
      DENOM=2.0*(XN+ZC**2)
      PM=(2.0*XN*P+ZC**2)/DENOM
      PI=(ZC*SQRT(4.0*N*P*(1-P)+ZC**2))/DENOM
10    RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | CONFID | 1 | 4 10 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 41 | DENOM | REAL | | | REFS | 8 | 9 | DEFINED | 7 | |
| 0 | N | INTEGER | | F.P. | REFS | 4 | 5 | 9 | DEFINED | 1 |
| 0 | NS | INTEGER | | F.P. | REFS | 6 | DEFINED | 1 | | |
| 40 | P | REAL | | | REFS | 8 | 2*9 | DEFINED | 6 | |
| 0 | PI | REAL | | F.P. | DEFINED | 1 | 3 | 9 | | |
| 0 | PM | REAL | | F.P. | DEFINED | 1 | 2 | 8 | | |
| 37 | XN | REAL | | | REFS | 6 | 7 | 8 | DEFINED | 5 |
| 0 | ZC | REAL | | F.P. | REFS | 7 | 8 | 2*9 | DEFINED | 1 |

| EXTERNALS | TYPE | ARGS | | REFERENCES |
|---|---|---|---|---|
| SQRT | REAL | 1 | LIBRARY | 9 |

| INLINE FUNCTIONS | TYPE | ARGS | | DEF LINE | REFERENCES |
|---|---|---|---|---|---|
| FLOAT | REAL | 1 | INTRIN | | 5 |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 428 | 34 |

```
      SUBROUTINE CLEAR(NC,IVEC)
      DIMENSION IVEC(NC)
      DO 10 I=1,NC
        IVEC(I)=0
10    CONTINUE
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | CLEAR | 1 | 6 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 22 | I | INTEGER | | | | PEFS | 4 | DEFINED | 3 | |
| 0 | IVEC | INTEGER | ARRAY | F.P. | | REFS | 2 | DEFINED | 1 | 4 |
| 0 | NC | INTEGER | | F.P. | | REFS | 2 | 3 | DEFINED | 1 |

| STATEMENT LABELS | | DEF LINE | REFEPENCFS |
|---|---|---|---|
| 0 | 10 | 5 | 3 |

| LOOPS | LABEL | INDFX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|
| 16 | 10 | I | 3 5 | 28 | INSTACK |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 308 | 24 |

```
      SUBROUTINE COPY(TIME,DETEC,REASON,NEXT)
C                        THIS VERSION:   MARCH 1976
C IT COPIES THE NEW FAULT IN ACFAU. IF THERE IS TOO MANY FAULTS, REASON
C IS SET TO 1 AND NEXT TO  5
C
C ******************************************************************************
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,T
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/CO12/NTR,IFAU
      COMMON/CO15/NOON(5),NWORK
      COMMON/CO41/ICATAS,I3
      COMMON/CO60/LSTFLT
      COMMON/FCOUNT/NF(5),NTRF(5)
      INTEGER EXTEN,PTR,REASON
C********** ********* ********* ********** ********* ********* ********
C
      IFAU=IFAU+1
      IF (TABLE(PTR,2).LT.ENDMIS) NTR=NTR+1
      IF (NWORK.GE.3)  I3=I3+1
      DO 4 I=1,IDIM
      IF (ACFAU(I,3).NE.0.)GO TO 4
      LSTFLT=I
      ACFAU(I,1)=TIME
      ACFAU(I,2)=TIME+TABLE(PTR,2)
C        HERE WE COUNT OUR TRANSIENTS                                  SP12APR
      ACFAU(I,3)=TABLE(PTR,3)
      ACFAU(I,4)=TABLE(PTR,4)
      ACFAU(I,5)=EXTEN
      ACFAU(I,6)=DETEC
      RETURN
C
C THERE ARE MORE THAN 5 FAULTS.
    4 CONTINUE
      REASON=1
      NEXT=5
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | COPY | 1 | 30 36 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 8 | 21 | DEFINED | 23 | 24 | 26 | 27 |
| | | | | | | 28 | 29 | | | | | |
| 2265 | DELAY | REAL | | COM1 | REFS | 7 | | | | | | |
| 0 | DETEC | REAL | | F.P. | REFS | 29 | DEFINED | 1 | | | | |
| 74 | ENDMIS | REAL | | COM3 | REFS | 8 | 18 | | | | | |
| 2262 | EXTEN | INTEGER | | COM1 | REFS | 7 | 14 | 28 | | | | |
| 56 | I | INTEGER | | | REFS | 21 | 22 | 23 | 24 | 26 | 27 | 28 |
| | | | | | | 29 | DEFINED | 20 | | | | |
| 0 | ICATAS | INTEGER | | C041 | REFS | 11 | | | | | | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 7 | | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 7 | 20 | | | | | |
| 1 | IFAU | INTEGER | | C012 | REFS | 9 | 17 | DEFINED | 17 | | | |
| 1 | I3 | INTEGER | | C041 | REFS | 11 | 19 | DEFINED | 19 | | | |
| 0 | LSTFLT | INTEGER | | C060 | REFS | 12 | DEFINED | 22 | | | | |
| 75 | MEMSIZ | INTEGER | | COM3 | REFS | 8 | | | | | | |
| 0 | NEXT | INTEGER | | F.P. | DEFINED | 1 | 35 | | | | | |
| 0 | NF | INTEGER | ARRAY | FCOUNT | REFS | 13 | | | | | | |
| 0 | NOON | INTEGER | ARRAY | C015 | REFS | 10 | | | | | | |
| 0 | NTR | INTEGER | | C012 | REFS | 9 | 18 | DEFINED | 18 | | | |
| 5 | NTRF | INTEGER | ARRAY | FCOUNT | REFS | 13 | | | | | | |
| 5 | NWORK | INTEGER | | C015 | REFS | 10 | 19 | | | | | |
| 2261 | PTR | INTEGER | | COM1 | REFS | 7 | 14 | 18 | 24 | 26 | 27 | |
| 0 | REASON | INTEGER | | F.P. | REFS | 14 | DEFINED | 1 | 34 | | | |
| 2264 | RECOV | REAL | | COM1 | REFS | 7 | | | | | | |
| 2266 | T | REAL | | COM1 | REFS | 7 | | | | | | |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 7 | 18 | 24 | 26 | 27 | | |
| 76 | TC | REAL | | COM3 | REFS | 8 | | | | | | |
| 0 | TIME | REAL | | F.P. | REFS | 23 | 24 | DEFINED | 1 | | | |

| STATEMENT LABELS | DEF LINE | REFERENCES |
|---|---|---|
| 47 4 | 33 | 20 21 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | EXITS |
|---|---|---|---|---|---|---|
| 34 | 4 | * I | 20 33 | 158 | OPT | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 IDIM (1) | 1 TABLE (1200) | 1201 PTR (1) | | | |
| | | 1202 EXTEN (1) | 1203 IDETEC (1) | 1204 RECOV (1) | | | |
| | | 1205 DELAY (1) | 1206 T (1) | | | | |
| COM3 | 63 | 0 ACFAU (60) | 60 ENDMIS (1) | 61 MEMSIZ (1) | | | |
| | | 62 TC (1) | | | | | |
| C012 | 2 | 0 NTR (1) | 1 IFAU (1) | | | | |
| C015 | 6 | 0 NOON (5) | 5 NWORK (1) | | | | |
| C041 | 2 | 0 ICATAS (1) | 1 I3 (1) | | | | |
| C060 | 1 | 0 LSTFLT (1) | | | | | |
| FCOUNT | 10 | 0 NF (5) | 5 NTRF (5) | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 678 | 55 |
| COMMON LENGTH | 24138 | 1291 |

```
      SUBROUTINE  DDUFLT(DDUNO,DEV)
C
C DETERMINES THE EFFECT OF A DDU DEVICE FAILURE ON THE DPS SUBSYSTEM
C
C DDUNO  THE DDU THAT CONTROLS THE DEVICE WHERE THE FAULT OCCURS
C DEV    THE DEVICE WHERE THE FAULT OCCURS
C
C*************************************************************************
      INTEGER DDUNO,DEV
      COMMON/FCB7/NDDUDV,DDUDVS(4),DDUDV(2,4),DDUDVN(4)
      COMMON/FCB11/NDDUTF(4),NDDUPF(4),NDDUTR(4)
      COMMON/FCB12/DDUCOV(4),DDUTFL(4),DDUTFD(4)
      COMMON/FAULT/FLTTYP
      COMMON/C012/NTR,IFAU
      COMMON/STATUS/STS(20)
      COMMON/FCBUC/FCBUCF(6)
      COMMON/FLAGS/FLG(5)
      LOGICAL FLG
      INTEGER FCBUCF
      LOGICAL GREATR
      INTEGER DDUDVS,DDUDV,DDUDVN,FLTTYP,STS
      LOGICAL LESS
C               ...RETURN IF THE DEVICE HAS ALREADY FAILED
      IF(DDUDV(DDUNO,DEV).EQ.0) RETURN
      FLG(2)=.TRUE.
      IFAU = IFAU+1
      GOTO(100,200),FLTTYP
C
C               ...THE FAULT IS TRANSIENT
  100 CONTINUE
      NTR=NTR+1
      NDDUTF(DEV)=NDDUTF(DEV)+1
      NL=DDUDVS(DEV)
      IF(GREATR(DDUTFD(DEV))) GOTO 240
      IF(GREATR(DDUTFL(DEV))) RETURN
      NDDUTR(DEV)=NDDUTR(DEV)+1
      GOTO 205
C
C               ...THE FAULT IS PERMANENT
  200 CONTINUE
      NDDUPF(DEV)=NDDUPF(DEV)+1
C               ...ENTER LEAKY TRANSIENTS
  205 CONTINUE
      DDUDV(DDUNO,DEV)=0
      NL=DDUDVS(DEV)-1
      DDUDVS(DEV)=NL
      IF(NL.EQ.0) GOTO 250
      IF(LESS(DDUCOV(DEV))) RETURN
  240 CONTINUE
C               ...AN UNCOVERED FAILURE
      CALL SETSTS(FCBUCF(3))
      RETURN
C
C               ...SYSTEM  FAILURE
  250 CONTINUE
```

```
      CALL SETSTS(DDUOVN(DEV))
      RETURN
      END
```

SYMBOLIC REFEPENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | | | |
|---|---|---|---|---|---|---|---|
| 2 | DDUFLT | 1 | 24 | 35 | 48 | 52 | 57 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | DDUCOV | REAL | ARRAY | FCB12 | REFS | 12 | 48 | | | | |
| 5 | DDUDV | INTEGER | ARRAY | FCB7 | REFS | 10 | 21 | 24 | DEFINED | 44 | |
| 15 | DDUDVN | INTEGER | ARRAY | FCB7 | REFS | 10 | 21 | 56 | | | |
| 1 | DDUDVS | INTEGER | ARRAY | FCB7 | REFS | 10 | 21 | 33 | 45 | DEFINED | 46 |
| 0 | DDUNO | INTEGER | | F.P. | REFS | 9 | 24 | 44 | DEFINED | 1 | |
| 10 | DDUTFD | REAL | ARRAY | FCB12 | REFS | 12 | 34 | | | | |
| 4 | DDUTFL | REAL | ARRAY | FCB12 | REFS | 12 | 35 | | | | |
| 0 | DEV | INTEGER | | F.P. | REFS | 9 | 24 | 2*32 | 33 | 34 | 35 | 2*36 |
| | | | | | 2*41 | 44 | 45 | 46 | 48 | 56 | | |
| | | | | | DEFINED | 1 | | | | | | |
| 0 | FCBUCF | INTEGER | ARRAY | FCBUC | REFS | 16 | 19 | 51 | | | |
| 0 | FLG | LOGICAL | ARRAY | FLAGS | REFS | 17 | 18 | DEFINED | 25 | | |
| 0 | FLTTYP | INTEGER | | FAULT | REFS | 13 | 21 | 27 | | | |
| 1 | IFAU | INTEGER | | C012 | REFS | 14 | 26 | DEFINED | 26 | | |
| 0 | NDDUDV | INTEGER | | FCB7 | REFS | 10 | | | | | |
| 4 | NDDUPF | INTEGER | ARRAY | FCB11 | REFS | 11 | 41 | DEFINED | 41 | | |
| 0 | NDDUTF | INTEGER | ARRAY | FCB11 | REFS | 11 | 32 | DEFINED | 32 | | |
| 10 | NDDUTR | INTEGER | ARRAY | FCB11 | REFS | 11 | 36 | DEFINED | 36 | | |
| 101 | NL | INTEGER | | | REFS | 46 | 47 | DEFINED | 33 | 45 | |
| 0 | NTR | INTEGER | | C012 | REFS | 14 | 31 | DEFINED | 31 | | |
| 0 | STS | INTEGER | ARRAY | STATUS | REFS | 15 | 21 | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES | | |
|---|---|---|---|---|---|
| GREATR | LOGICAL | 1 | 20 | 34 | 35 |
| LESS | LOGICAL | 1 | 22 | 48 | |
| SETSTS | | 1 | 51 | 56 | |

| STATEMENT LABELS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 23 | 100 | 30 | 27 |
| 44 | 200 | 40 | 27 |
| 46 | 205 | 43 | 37 |
| 62 | 240 | 49 | 34 |
| 66 | 250 | 55 | 47 |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | |
|---|---|---|---|---|---|
| FCB7 | 17 | 0 NDDUDV (1) | | 1 DDUDVS (4) | 5 DDUDV (8) |
| | | 13 DDUDVN (4) | | | |
| FCB11 | 12 | 0 NDDUTF (4) | | 4 NDDUPF (4) | 8 NDDUTR (4) |
| FCB12 | 12 | 0 DDUCOV (4) | | 4 DDUTFL (4) | 8 DDUTFD (4) |
| FAULT | 1 | 0 FLTTYP (1) | | | |
| C012 | 2 | 0 NTR (1) | | 1 IFAU (1) | |
| STATUS | 20 | 0 STS (20) | | | |
| FCBUC | 6 | 0 FCBUCF (6) | | | |
| FLAGS | 5 | 0 FLG (5) | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 1028 | 66 |
| COMMON LENGTH | 1138 | 75 |

Q2

```
      SUBROUTINE DETTIM(DETEC,TIME,PLACE)
C                        THIS VERSION:   MARCH 1976
C THIS SUBROUTINE DETERMINES THE  DETECTION TIME OF THE FAULT DESCRIBED
C BY TIME AND PLACE: TIME IS THE PSEUDO-OCCURRENCETIME, PLACE THE PSEUDO
C MODULE. THE TIME OF THE COMPARISON FOLLOWING OCCURRENCE OF THE FAULT
C IS FIRST DETERMINED. MEMORY FAULT MAY BE DETECTED LATER.
C DETEC   DETECTION TIME COMPUTED BY DETTIME
C TIME    OCCURRENCE TIME OF THE FAULT
C PLACE   MODULE WHERE THE FAULT OCCURS
C
C********** ********* ********* ********** ********* ********* *********
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/CO31/RATINT,ISYNC
      COMMON/DETE/PDET,DETMAX,PDM
      COMMON/EXEC/MINCY,RTI,TODO,OLTIS,IDLE,SEQMAX,DOMMAX,MISITE
      COMMON/MM/PROMM,MACY
C ***************************************************************************
C
      IF (ISYNC.EQ.0) GO TO 130
C
C SYNCHRONOUS SCHEDULING
      DETEC=TC*AINT(TIME/TC)+TC
      IF ((PLACE.NE.3.).OR.(RANF(0.).LT.PDM)) RETURN
      IF (RANF(0.).GT.PROMM)GO TO 110
      NCOM=RTI/TC
      GO TO 120
  110 CONTINUE
      NCOM=(MACY*RTI)/TC
  120 CONTINUE
      N=IRAN(1,NCOM)
      DETEC=DETEC+(N-1)*TC
      RETURN
C
C ASYNCHRONOUS SCHEDULING
  130 CONTINUE
      U=RANF(0.)
      IF ((PLACE.EQ.3.).AND.(U.GT.PDM)) GO TO 135
      DETEC=TIME+TC*2.*RANF(0.)
      RETURN
  135 CONTINUE
      IF (RANF(0.).GT.PROMM) GO TO 140
      DET=RTI
      GO TO 150
  140 CONTINUE
      DET=RTI*MACY
  150 CONTINUE
      DETEC=TIME+DET*2.*RANF(0.)
      END
```

SYMBOLIC REFERENCE MAP

ENTRY POINTS DEF LINE REFERENCES

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | DETTIM | 1 | 23 | 32 | 39 | 48 |

VARIABLES SN TYPE RELOCATION

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 12 | | | | | | |
| 116 | DET | REAL | | | REFS | 47 | DEFINED | 42 | 45 | | | |
| 0 | DETEC | REAL | | F.P. | REFS | 31 | DEFINED | 1 | 22 | 31 | 38 | 47 |
| 1 | DETMAX | REAL | | DETE | REFS | 14 | | | | | | |
| 6 | DOWMAX | REAL | | EXEC | REFS | 15 | | | | | | |
| 74 | ENDMIS | REAL | | COM3 | REFS | 12 | | | | | | |
| 4 | IDLE | INTEGER | | EXEC | REFS | 15 | | | | | | |
| 1 | ISYNC | INTEGER | | C031 | REFS | 13 | 19 | | | | | |
| 1 | MACY | INTEGER | | MM | REFS | 16 | 28 | 45 | | | | |
| 75 | MEMSIZ | INTEGER | | COM3 | REFS | 12 | | | | | | |
| 0 | MINCY | INTEGER | | EXEC | REFS | 15 | | | | | | |
| 7 | MISITE | INTEGER | | EXEC | REFS | 15 | | | | | | |
| 114 | N | INTEGER | | | REFS | 31 | DEFINED | 30 | | | | |
| 113 | NCOM | INTEGER | | | REFS | 30 | DEFINED | 25 | 28 | | | |
| 3 | OLTIS | REAL | | EXEC | REFS | 15 | | | | | | |
| 0 | PDET | REAL | | DETE | REFS | 14 | | | | | | |
| 2 | PDM | REAL | | DETE | REFS | 14 | 23 | 37 | | | | |
| 0 | PLACE | REAL | | F.P. | REFS | 23 | 37 | DEFINED | 1 | | | |
| 0 | PROMM | REAL | | MM | REFS | 16 | 24 | 41 | | | | |
| 0 | RATINT | REAL | | C031 | REFS | 13 | | | | | | |
| 1 | RTI | REAL | | EXEC | REFS | 15 | 25 | 28 | 42 | 45 | | |
| 5 | SEQMAX | REAL | | EXEC | REFS | 15 | | | | | | |
| 76 | TC | REAL | | COM3 | REFS | 12 | 3*22 | 25 | 28 | 31 | 38 | |
| 0 | TIME | REAL | | F.P. | REFS | 22 | 38 | 47 | DEFINED | 1 | | |
| 2 | TODO | REAL | | EXEC | REFS | 15 | | | | | | |
| 115 | U | REAL | | | REFS | 37 | DEFINED | 36 | | | | |

EXTERNALS TYPE ARGS REFERENCES

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| IRAN | INTEGER | 2 | 30 | | | | | |
| RANF | REAL | 1 | 23 | 24 | 36 | 38 | 41 | 47 |

INLINE FUNCTIONS TYPE ARGS DEF LINE REFERENCES

| | | | | | |
|---|---|---|---|---|---|
| AINT | REAL | 1 | INTRIN | 22 | |

STATEMENT LABELS DEF LINE REFERENCES

| | | | |
|---|---|---|---|
| 31 | 110 | 27 | 24 |
| 35 | 120 | 29 | 26 |
| 44 | 130 | 35 | 19 |
| 62 | 135 | 40 | 37 |
| 70 | 140 | 44 | 41 |
| 73 | 150 | 46 | 43 |

COMMON BLOCKS LENGTH MEMBERS - BIAS NAME(LENGTH)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COM3 | 63 | 0 | ACFAU | (60) | 60 | ENDMIS | (1) | 61 MEMSIZ (1) |
| | | 62 | TC | (1) | | | | |
| C031 | 2 | 0 | RATINT | (1) | 1 | ISYNC | (1) | |
| DETE | 3 | 0 | PDET | (1) | 1 | DETMAX | (1) | 2 PDM (1) |
| EXEC | 8 | 0 | MINCY | (1) | 1 | RTI | (1) | 2 TODO (1) |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | |
|---|---|---|---|---|
| | | 3 OLTIS (1) | 4 IDLE (1) | 5 SEQMAX (1) |
| | | 6 DOWMAX (1) | 7 MISITE (1) | |
| MM | 2 | 0 PROMM (1) | 1 MACY (1) | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 117B | 79 |
| COMMON LENGTH | 116B | 78 |

```
      SUBROUTINE DFAFLT(MDM,DEV)
C
C THIS ROUTINE DETERMINES THE EFFECT OF A FAULT IN ONE OF THE DEVICES
C INTERFACED TO THE FLIGHT-AFT MDMS.
C
C MDM    IDENTIFIES THE FA-MDM INTERFACED TO THE BAD DEVICE
C DEV    IDENTIFIES THE DEVICE
C
C*********************************************************************
      COMMON/FAULT/FLTTYP
      COMMON/C012/NTR,IFAU
      COMMON/STATUS/STS(20)
      COMMON/FCB10/NDFADV,DFADVS(3),DFADV(4,3),DFADVN(3)
      COMMON/FCB17/NDFATF(3),NDFAPF(3),NDFATR(3)
      COMMON/FCB18/DFATFD(3),DFATFL(3),DFACOV(3)
      COMMON/FLAGS/FLG(5)
      COMMON/FCBUC/FCBUCF(6)
      LOGICAL FLG
      INTEGER FCBUCF
      LOGICAL GREATR
      INTEGER DEV,FLTTYP,STS,DFADVS,DFADV,DFADVN
      LOGICAL LESS
C               ...RETURN IF UNIT HAS ALREADY FAILED
      IF(DFADV(MDM,DEV).EQ.0) RETURN
      FLG(5)=.TRUE.
      IFAU=IFAU+1
      GOTO(100,200),FLTTYP
C
C               ...A TRANSIENT HAS OCCURRED
  100 CONTINUE
      NTR=NTR+1
      NDFATF(DEV)=NDFATF(DEV)+1
      NL=DFADVS(DEV)
      IF((DEV.EQ.1).AND.(NL.LE.2)) GOTO 250
      IF(GREATR(DFATFD(DEV)).AND.(NL.EQ.1)) GOTO 800
      IF(GREATR(DFATFL(DEV))) RETURN
      NDFATR(DEV)=NDFATR(DEV)+1
      GOTO 205
C
C               ...ITS A PERMANENT FAULT
  200 CONTINUE
      NDFAPF(DEV)=NDFAPF(DEV)+1
C               ...ENTER LEAKY TRANSIENTS---RECONFIGURATION
  205 CONTINUE
      DFADV(MDM,DEV)=0
      NL=DFADVS(DEV)-1
      DFADVS(DEV)=NL
      IF(NL.EQ.0) GOTO 250
      IF(NL.GT.1) RETURN
      IF(LESS(DFACOV(DEV)))RETURN
C
C               ...A SET OF DEVICES IS NO LONGER FUNCTIONAL
  250 CONTINUE
      CALL SETSTS(DFADVN(DEV))
      RETURN
```

```
      C                ...AN UNCOVERED(CATESTROPHIC) FAULT OCCURRED
        800 CONTINUE
            CALL SETSTS(FCBUCF(6))
            RETURN
60          END
```

SYMBOLIC REFERENCE MAP

```
ENTRY POINTS     DEF LINE     REFERENCES
    2  DFAFLT          1        24        36        49        50        55        59

VARIABLES     SN  TYPE            RELOCATION
    0  DEV        INTEGER               F.P.        REFS      21        24      2*32        33        34        35        36
                                                     2*37    2*42        45        46        47        50        54
                                                  DEFINED       1
    6  DFACOV     REAL        ARRAY     FCB18       REFS      15        50
    4  DFADV      INTEGER     ARRAY     FCB10       REFS      13        21        24   DEFINED        45
   20  DFADVN     INTEGER     ARRAY     FCB10       REFS      13        21        54
    1  DFADVS     INTEGER     ARRAY     FCB10       REFS      13        21        33        46   DEFINED        47
    0  DFATFD     REAL        ARRAY     FCB18       REFS      15        35
    3  DFATFL     REAL        ARRAY     FCB18       REFS      15        36
    0  FCBUCF     INTEGER     ARRAY     FCBUC       REFS      17        19        58
    0  FLG        LOGICAL     ARRAY     FLAGS       REFS      16        18   DEFINED        25
    0  FLTTYP     INTEGER               FAULT       REFS      10        21        27
    1  IFAU       INTEGER               C012        REFS      11        26   DEFINED        26
    0  MDM        INTEGER                F.P.       REFS      24        45   DEFINED         1
    0  NDFADV     INTEGER               FCB10       REFS      13
    3  NDFAPF     INTEGER     ARRAY     FCB17       REFS      14        42   DEFINED        42
    0  NDFATF     INTEGER     ARRAY     FCB17       REFS      14        32   DEFINED        32
    6  NDFATR     INTEGER     ARRAY     FCB17       REFS      14        37   DEFINED        37
  115  NL         INTEGER                           REFS      34        35        47        48        49
                                                  DEFINED      33        46
    0  NTR        INTEGER               C012        REFS      11        31   DEFINED        31
    0  STS        INTEGER     ARRAY     STATUS      REFS      12        21

EXTERNALS          TYPE   ARGS      REFERENCES
       GREATR      LOGICAL   1           20        35        36
       LESS        LOGICAL   1           22        50
       SETSTS                1           54        58

STATEMENT LABELS              DEF LINE    REFERENCES
   23  100                        30         27
   53  200                        41         27
   55  205                        44         38
   76  250                        53         34        48
  104  800                        57         35

COMMON BLOCKS   LENGTH     MEMBERS - BIAS NAME(LENGTH)
       FAULT         1              0 FLTTYP (1)
       C012          2              0 NTR    (1)              1 IFAU   (1)
       STATUS       20              0 STS    (20)
       FCB10        19              0 NDFADV (1)              1 DFADVS (3)              4 DFADV  (12)
                                   16 DFADVN (3)
       FCB17         9              0 NDFATF (3)              3 NDFAPF (3)              6 NDFATR (3)
       FCB18         9              0 DFATFD (3)              3 DFATFL (3)              6 DFACOV (3)
       FLAGS         5              0 FLG    (5)
       FCBUC         6              0 FCBUCF (6)
```

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 1168 | 78 |
| COMMON LENGTH | 1078 | 71 |

```
      SUBROUTINE  DFFFLT(MDMN0,DEV)
C
C     DETERMINES THE EFFECT OF A FLIGHT FORWARD MDM DEDICATED DEVICE
C     FAULT.
C
C MDMN0  IDENTIFIES THE FF-MDM THAT IS INTERFACED WITH THE BAD DEVICE
C DEV    IDENTIFIES THE TYPE OF DEVICE THAT FAILED
C
C*****************************************************************
      COMMON/FCB8/NDFFDV,DFFDVS(6),DFFDV(4,6),DFFDVN(6)
      COMMON/FCB13/NDFFTF(6),NDFFPF(6),NDFFTR(6)
      COMMON/FCB14/DFFTFD(6,4),DFFTFL(6),DFFCDV(6,3)
      COMMON/FAULT/FLTTYP
      COMMON/CO12/NTR,IFAU
      COMMON/STATUS/STS(20)
      COMMON/FCBUC/FCBUCF(6)
      COMMON/FLAGS/FLG(5)
      LOGICAL FLG
      LOGICAL GREATR
      INTEGER FCBUCF
      INTEGER DEV,DFFDVS,DFFDV,DFFDVN,FLTTYP,STS
      LOGICAL LESS
C
C                 ...RETURN IF UNIT HAS ALREADY FAILED
      IF(DFFDV(MDMN0,DEV).EQ.0) RETURN
      IFAU=IFAU+1
      FLG(3)=.TRUE.
      GOTO(100,200),FLTTYP
C
C                 ...A TRANSIENT HAS OCCURRED
  100 CONTINUE
      NTR=NTR+1
      NDFFTF(DEV)=NDFFTF(DEV)+1
C                 ...IS THE FAULT DETECTED
      NL=DFFDVS(DEV)
      IF(GREATR(DFFTFD(DEV,NL))) GOTO 800
C                 ...IS TRANSIENT RECOVERY SUCCESSFUL
      IF(GREATR(DFFTFL(DEV))) RETURN
      NDFFTR(DEV) = NDFFTR(DEV)+1
      GOTO 205
C
C                 ...A PERMANENT FAULT HAS OCCURED
  200 CONTINUE
      NDFFPF(DEV)=NDFFPF(DEV)+1
      NL=DFFDVS(DEV)
C                 ...ENTER  LEAKY TRANSIENTS
  205 CONTINUE
      DFFDV(MDMN0,DEV)=0
      NL=DFFDVS(DEV)-1
      DFFDVS(DEV)=NL
      IF(NL.EQ.0) GOTO 250
      IF(LESS(DFFCDV(DEV,NL))) RETURN
C                 ...THE DEVICE GROUP IS NO LONGER FUNCTIONAL
  250 CONTINUE
      CALL SETSTS(DFFDVN(DEV))
```

```
      RETURN
C
C                ...THE FAULT WAS UNDETECTED
  800 CONTINUE
60    CALL SETSTS(FCBUCF(4))
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | DEF LINE | REFERENCES | | | | | |
|---|---|---|---|---|---|---|---|
| 2 OFFFLT | 1 | 25 | 38 | 52 | 56 | 61 | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | DFV | INTEGER | | F.P. | REFS | 21 | 25 | 2*33 | 35 | 36 | 38 | 2*39 |
| | | | | | 2*44 | 45 | 48 | 49 | 50 | 52 | 55 | |
| | | | | | DEFINED | 1 | | | | | | |
| 36 | OFFCOV | REAL | ARRAY | FCB14 | REFS | 12 | 52 | | | | | |
| 7 | OFFDV | INTEGER | ARRAY | FCB8 | REFS | 10 | 21 | 25 | DEFINED | 48 | | |
| 37 | OFFDVN | INTEGER | ARRAY | FCB8 | REFS | 10 | 21 | 55 | | | | |
| 1 | OFFDVS | INTEGER | ARRAY | FCB8 | REFS | 10 | 21 | 35 | 45 | 49 | | |
| | | | | | DEFINED | 50 | | | | | | |
| 0 | OFFTFD | REAL | ARRAY | FCB14 | REFS | 12 | 36 | | | | | |
| 30 | OFFTFL | REAL | ARRAY | FCB14 | REFS | 12 | 38 | | | | | |
| 0 | FCBUCF | INTEGER | ARRAY | FCBUC | REFS | 16 | 20 | 60 | | | | |
| 0 | FLG | LOGICAL | ARRAY | FLAGS | REFS | 17 | 18 | DEFINED | 27 | | | |
| 0 | FLTTYP | INTEGER | | FAULT | REFS | 13 | 21 | 28 | | | | |
| 1 | IFAU | INTEGER | | C012 | REFS | 14 | 26 | DEFINED | 26 | | | |
| 0 | MDMNO | INTEGER | | F.P. | REFS | 25 | 48 | DEFINED | 1 | | | |
| 0 | NOFFDV | INTEGER | | FCB8 | REFS | 10 | | | | | | |
| 6 | NOFFPF | INTEGER | ARRAY | FCB13 | REFS | 11 | 44 | DEFINED | 44 | | | |
| 0 | NOFFTF | INTEGER | ARRAY | FCB13 | REFS | 11 | 33 | DEFINED | 33 | | | |
| 14 | NOFFTR | INTEGER | ARRAY | FCB13 | REFS | 11 | 39 | DEFINED | 39 | | | |
| 105 | NL | INTEGER | | | REFS | 36 | 50 | 51 | 52 | DEFINED | 35 | 45 |
| | | | | | 49 | | | | | | | |
| 0 | NTR | INTEGER | | C012 | REFS | 14 | 32 | DEFINED | 32 | | | |
| 0 | STS | INTEGER | ARRAY | STATUS | REFS | 15 | 21 | | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES | | | |
|---|---|---|---|---|---|---|
| GREATR | LOGICAL | 1 | 19 | 36 | 38 | |
| LESS | LOGICAL | 1 | 22 | 52 | | |
| SETSTS | | 1 | 55 | 60 | | |

| STATEMENT LABELS | DEF LINE | REFERENCES |
|---|---|---|
| 23 100 | 31 | 28 |
| 44 200 | 43 | 28 |
| 50 205 | 47 | 40 |
| 66 250 | 54 | 51 |
| 74 800 | 59 | 36 |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | |
|---|---|---|---|---|
| FCB8 | 37 | 0 NOFFDV (1) | 1 OFFDVS (6) | 7 OFFDV (24) |
| | | 31 OFFDVN (6) | | |
| FCB13 | 18 | 0 NOFFTF (6) | 6 NOFFPF (6) | 12 NOFFTR (6) |
| FCB14 | 48 | 0 OFFTFD (24) | 24 OFFTFL (6) | 30 OFFCOV (18) |
| FAULT | 1 | 0 FLTTYP (1) | | |
| C012 | 2 | 0 NTR (1) | 1 IFAU (1) | |
| STATUS | 20 | 0 STS (20) | | |
| FCBUC | 6 | 0 FCBUCF (6) | | |
| FLAGS | 5 | 0 FLG (5) | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 106B | 70 |
| COMMON LENGTH | 211B | 137 |

```
      SUBROUTINE EXDUR(FAU,N,DUR)
C                                        THIS VERSION: 25 FEBRUARY 1974
C  IT GENERATES N EXPONENTIAL DURATIONS
C FAU     ARRAY WHERE DURATIONS ARE STORED
C N       DIMENSION OF FAU
C DUR     MEAN DURATION
C
C********** ********* ********* ********** ********* ********* ********
      DIMENSION FAU(N)
C ***************************************************************************
C
      DO 40 K=1,N
   41 U=RANF(0.)
      IF (U.EQ.0.)GO TO 41
      FAU(K)=-DUR*ALOG(U)
   40 CONTINUE
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | EXDUR | 1 | 17 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | DUR | REAL | | F.P. | REFS | 15 | DEFINED | 1 | | |
| 0 | FAU | REAL | ARRAY | F.P. | REFS | 9 | DEFINED | 1 | 15 | |
| 34 | K | INTEGER | | | REFS | 15 | DEFINED | 12 | | |
| 0 | N | INTEGER | | F.P. | REFS | 9 | 12 | DEFINED | 1 | |
| 35 | U | REAL | | | REFS | 14 | 15 | DEFINED | 13 | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| ALOG | REAL | 1 LIBRARY | 15 |
| RANF | REAL | 1 | 13 |

| STATEMENT LABELS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 0 | 40 | 16 | 12 |
| 15 | 41 | 13 | 14 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|
| 15 | 40 | * K | 12 16 | 138 | EXT REFS |

STATISTICS

PROGRAM LENGTH 44B 36

```
      SUBROUTINE EXPON(FAU,N,RTIME,NACTIV,RLAMDA)
C                          THIS VERSION:   MARCH 1976
C IT GENERATES N OR LESS POISSON ARRIVAL TIMES. GENERATION STOPS WHEN
C RTIME IS REACHED.
C FAU      ARRAY OF ARRIVAL TIMES
C N        DIMENSION OF FAU
C RTIME    UPPER LIMIT ON ARRIVAL TIME
C NACTIVE NUMBER OF UNITS RECEIVING FAULTS AT RATE RLAMBDA
C RLAMBDA FAULT RATE IN EACH UNIT
C
C********** ********* ********* ********** ********* ********* ********
      DIMENSION FAU(N)
C ***************************************************************************
C
      TIME=0.
      RLAMBD=NACTIV*RLAMDA/3.6E6
      IF (RLAMBD.EQ.0.) GO TO 19
      DO 17 K=1,N
  170 CONTINUE
      U=RANF(0.)
      IF (U.EQ.0.) GO TO 170
      Y=(-1./RLAMBD)*ALOG(U)
      TIME=Y+TIME
      FAU(K)=TIME
      IF(TIME.GT.RTIME)GOTO 18
   17 CONTINUE
   20 CONTINUE
      FAU(N)=RTIME
      RETURN
   18 CONTINUE
      N=K
      RETURN
   19 CONTINUF
      N=1
      GO TO 20
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | EXPON | 1 | 29 32 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | FAU | REAL | ARRAY | F.P. | REFS | 12 | DEFINED | 1 | 24 | 28 | |
| 63 | K | INTEGER | | | REFS | 24 | 31 | DEFINED | 18 | | |
| 0 | N | INTEGER | | F.P. | REFS | 12 | 18 | 26 | DEFINED | 1 | 31 34 |
| 0 | NACTIV | INTEGER | | F.P. | REFS | 16 | DEFINED | 1 | | | |
| 62 | RLAMBD | REAL | | | REFS | 17 | 22 | DEFINED | 16 | | |
| 0 | RLAMDA | REAL | | F.P. | REFS | 16 | DEFINED | 1 | | | |
| 0 | RTIME | REAL | | F.P. | REFS | 25 | 28 | DEFINED | 1 | | |
| 61 | TIME | REAL | | | REFS | 23 | 24 | 25 | DEFINED | 15 | 23 |
| 64 | U | REAL | | | REFS | 21 | 22 | DEFINED | 20 | | |
| 65 | Y | REAL | | | REFS | 23 | DEFINED | 22 | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| ALOG | REAL | 1 LIBRARY | 22 |
| RANF | REAL | 1 | 20 |

| STATEMENT LABELS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 0 | 17 | 26 | 18 |
| 46 | 18 | 30 | 25 |
| 51 | 19 | 33 | 17 |
| 42 | 20 | 27 | 35 |
| 23 | 170 | 19 | 21 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | EXT REFS | EXITS |
|---|---|---|---|---|---|---|---|
| 23 | 17 | * K | 18 26 | 170 | | | |

STATISTICS

PROGRAM LENGTH 101B 65

```
      SUBROUTINE EXTENT(EXT,DUR,MOD)
C                                     THIS VERSION: 18 JULY 1974
C THIS SUBROUTINES COMPUTES THE EXTENT OF A FAULT WHOSE DURATION IS DUR.
C DUR     UNUSED
C EXT     FAULT EXTENT: 0 NO MEMORY DAMAGE;1 MEMORY DAMAGE
C MOD     LOCATION OF THE FAULT
C
C ************************************************************************
      COMMON/PERM/LAST(5),MININT,PSUC
      INTEGER EXT
      REAL MOD
C********** ********* ********* ********** ********* ********* ********
C
      EXT=0
      IF (MOD.NE.3.)RETURN
      IF (RANF(0.) .GT.PSUC) EXT=1
      END
```

SYMBOLIC REFERENCE MAP

```
ENTRY POINTS      DEF LINE      REFERENCES
    2  EXTENT          1         15        17

VARIABLES     SN  TYPE          RELOCATION
    0  OUR        REAL       *UNUSED    F.P.       DEFINED        1
    0  EXT        INTEGER               F.P.         REFS        10    DEFINED        1        14        16
    0  LAST       INTEGER    ARRAY      PERM         REFS         9
    5  MININT     INTEGER               PERM         REFS         9
    0  MOD        REAL                  F.P.         REFS        11        15    DEFINED        1
    6  PSUC       REAL                  PERM         REFS         9        16

EXTERNALS           TYPE    ARGS       REFERENCES
       RANF        REAL       1             16

COMMON BLOCKS    LENGTH     MEMBERS - BIAS NAME(LENGTH)
       PERM           7               0 LAST   (5)                5 MININT (1)                6 PSUC   (1)

STATISTICS
  PROGRAM LENGTH      248       20
   COMMON LENGTH       78        7
```

```
      SUBROUTINE FAUGEN(NAC,RTIME,NFAUT,NMODU)
C                       THIS VERSION:  MARCH 1976
C FAUGENE GENERATES UP TO 300 FAULTS (NFAUT IS LESS THAN 300)
C THERE ARE UP TO 5 COMPUTERS(NACTIVE) WITH UP TO  5 MODULES (NMODU).
C THE 5 COMPUTERS ARE IDENTICAL. FOR EACH MODULE, THE COMPUTER IS
C RANDOMLY DETERMINED. NO MORE THAN 300 FAULTS PER MODULE.
C NO MORE THAN 150 PERMANENTS AND 150 TRANSIENTS.
C IN CASE OF BURST DISTRIBUTION, ONE HAS TO BE CAREFUL THAT THERE WILL
C NOT BE MORE THAN 10 FAULTS AT ANY TIME IN ACFAU. IF THIS SHOULD
C HAPPEN, INCREASE IDIM AND THE SIZE OF ACFAU.
C AVDUR(I) = AVERAGE DURATION OF THE TRANSIENTS (EXPONENTIAL DISTRIB.)
C BURST  = FAULT RATE DURING THE BURST
C DDUR(I) = 1 : UNIFORM DISTRIBUTION FOR DURATION
C DDUR(I) = 2 : EXPONENTIAL DISTRIBUTION FOR DURATION
C DPER(I) = 1 : POISSON DISTRIBUTION FOR ARRIVALS OF PERMANENTS
C DTRA(I) = 1 : POISSON DISTRIBUTION FOR ARRIVALS OF TRANSIENTS
C DTRA(I) = 2 : BURST DISTRIBUTION FOR ARRIVALS OF TRANSIENTS
C DURA   = DURATION OF THE BURST
C NAC    = NUMBER OF COMPUTERS
C NFAUT  = TOTAL NUMBER OF FAULTS
C RLAMB  = BURST OCCURRENCE RATE
C RMAXI(I) = MAXIMUM DURATION OF THE TRANSIENTS (UNIFORM DISTRIBUTION)
C RMINI(I) = MINIMUM DURATION OF THE TRANSIENTS (UNIFORM DISTRIBUTION)
C NMODU  = NUMBER OF DIFFERENT UNITS IN A COMPUTER
C RTIME  = MISSION TIME
C TABLE(I,1) = OCCURRENCE TIME
C TABLE(I,2) = DURATION (= RTIME IF PERMANENT)
C TABLE(I,3) = MODULE
C TABLE(I,4) = COMPUTER
C
C ***************************************************************
C
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTFN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM2/MIS,IFULL
      COMMON/CO14/RLAMDP(5),RLAMDT(5),RMINI(5),RMAXI(5),AVDUR(5)
      COMMON/CO28/MODSIM,NSPA
      COMMON/CO33/PMUP(5)
      COMMON/CO40/IIII,FFFF,NNNN,NIO
      COMMON/FLTMIS/NFCV(10),NSYSF(10),PFRIO,TFRIO
      INTEGER DPER(5),DTRA(5),DDUR(5)
      DIMENSION MERG1(5),MERG2(5)
      DIMENSION NA(5)
      DIMENSION RLAMDA(5),DURA(5),BURST(5)
      DIMENSION DFDDU(300)
      DIMENSION FAULTI(150,5,3),FAULTP(150,5),FAULT(150)
C ***************************************************************
C
C NMIS IS SET TO 1 WHEN MIS=0 SO THAT THE INPUT TEST WORKS
      NMIS=MIS+1
      DO 2 I=1,NMODU
      NPER=150
      NTRA=150
C SET NACTIVE TO THE NUMBER OF MODULES OF EACH KIND
      GO TO (5,4,5,16,5),I
    5 CONTINUE
```

```
                  NACTIV=NAC
                  GO TO 17
                8 CONTINUE
                  NACTIV=NTO
 60               GO TO 17
               16 CONTINUE
                  IF(NMIS.NE.1) GOTO 2016
                  CALL RDIOFR(PFRIO,TFRIO)
                  CALL PFCBCF
 65               PFRIO=PFRIO/1E6
                  TFRIO=TFRIO/1E6
             2016 CONTINUE
                  CALL EXPON(FAULT,NPER,RTIME,1,PFRIO)
                  IF(NPER.LT.150) GOTO 2010
 70                  IFULL=1
                     RETURN
             2010 CONTINUE
                  DO 2020 J=1,NPER
                     FAULTP(J,I)=FAULT(J)
 75          2020 CONTINUE
                  CALL EXPON(FAULT,NTRA,RTIME,1,TFRIO)
                  IF(NTRA.LT.150) GOTO 2030
                     IFULL=1
                     RETURN
 80          2030 CONTINUE
                  IF(NTRA.GT.1) GOTO 2035
                     FAULTT(1,I,1)=FAULT(1)
                     GOTO 2
             2035 CONTINUE
 85                DO 2040 J=1,NTRA
                     FAULTT(J,I,1)=FAULT(J)
                     CALL TFISO(IUNIT,1)
                     FAULTT(J,I,3)=IUNIT
                     FAULTT(J,I,2)=0.0
 90          2040 CONTINUE
                  GOTO 2
               17 CONTINUE
                  NA(I)=NACTIV
                  IF (NMIS.NE.1) GO TO 1
 95                READ 3,DPER(I),DTRA(I),DDUR(I)
                1 CONTINUE
                  K=DPER(I)
                  GO TO (4,6),K
                6 CONTINUE
100               K=DTRA(I)
                  GO TO (7,31,9),K
                9 CONTINUE
                  K=DDUR(I)
                  GO TO (10,30,2),K
105          C
             C GENERATION OF THE PERMANENTS (POISSON)
                4 CONTINUE
                  IF (NMIS.NE.1) GO TO 44
                  READ 105,RLAMDP(I)
110               RLAMDP(I)=PLAMDP(I)/1E6
```

```
                PRINT 990,I,RLAMDP(I)
                IF (NSPA.EQ.0) GO TO 44
                READ 105,RMUP(I)
                PRINT 997,RMUP(I)
115          44 CALL EXPON(FAULT,NPER,RTIME,NACTIV,RLAMDP(I))
                IF (NPER.LT.150) GO TO 14
                IFULL=1
                RETURN
             14 CONTINUE
120             DO 11 J=1,NPER
                FAULTP(J,I)=FAULT(J)
             11 CONTINUE
                GO TO 6
          C
125       C GENERATION OF THE TRANSIENTS (POISSON)
              7 CONTINUE
                IF (NMIS.NE.1) GO TO 77
                READ 105,RLAMDT(I)
                RLAMDT(I)=RLAMDT(I)/1E6
130             PRINT 980,RLAMDT(I)
             77 CALL EXPON(FAULT,NTRA,RTIME,NACTIV,RLAMDT(I))
                IF (NTRA.LT.150) GO TO 12
                IFULL=1
                RETURN
135          12 CONTINUE
                DO 13 J=1,NTRA
                FAULTT(J,I,1)=FAULT(J)
          C CHOOSE THE FAULTY COMPUTER
                FAULTT(J,I,3)=IRAN(1,NACTIV)
140          13 CONTINUE
                GOTO 9
          C
          C GENERATION OF THE BURSTS
             31 CONTINUE
145             IF (NMIS.NE.1) GO TO 32
                READ 999,RLAMB(I),DURA(I),BURST(I)
                RLAMB(I)=RLAMB(I)/1E6
                PRINT 998,RLAMB(I),DURA(I),BURST(I)
             32 CONTINUE
150             NTRC=150
                CALL EXPON(DEBBU,NTRC,RTIME,NACTIV,RLAMB(I))
                L=1
                DO 35 J=1,NTRC
                IF (DEBBU(J).GT.RTIME) GO TO 38
155          36 CONTINUE
                U=RANF(0.)
                IF (U.EQ.0.) GO TO 36
                U=-DURA(I)*ALOG(U)*1000.
          C WE AVOID A BURST COMING UPON A PREVIOUS BURST
160             U=AMIN1(U,DEBBU(J+1)-DEBBU(J))
                NTRB=150
                CALL EXPON(FAULT,NTRB,U,1,BURST(I)*3600.)
                M=IRAN(1,NACTIV)
                DO 37 K=1,NTRB
165             IF (FAULT(K).GT.U) GO TO 35
```

```
                FAULTT(L,I,1)=DERRU(J)*FAULT(K)
                FAULTT(L,I,3)=M
                L=L+1
                IF (L.GE.300) GO TO 38
  170        37 CONTINUE
             35 CONTINUE
             38 CONTINUE
                FAULTT(L,I,1)=10.*RTIME
                NTRA=L
  175           IF (L.LT.150) GO TO 9
                IFULL=1
                RETURN
           C
           C GENERATION OF THE TRANSIENT DURATIONS (UNIFORM)
  180        10 CONTINUE
                IF (NMIS.NE.1) GO TO 1010
                READ 19,RMINI(I),RMAXI(I)
                AVER=(RMAXI(I)+RMINI(I))/2
                PRINT 950,AVER
  185        1010 CONTINUE
           C IF NO TRANSIENT GO TO 2
                IF (NTRA.LE.1) GO TO 2
                CALL UNIF(FAULT,NTRA,RMINI(I),RMAXI(I))
                GO TO 1011
  190       C
           C GENERATION OF THE TRANSIENT DURATIONS (EXPONENTIAL)
             30 CONTINUE
                IF (NMIS.NE.1) GO TO 3030
                READ 105,AVDUR(I)
  195           PRINT 949,AVDUR(I)
           3030 CONTINUE
           C IF NO TRANSIENT GO TO 2
                IF (NTRA.LE.1) GO TO 2
                CALL EXDUR(FAULT,NTRA,AVDUR(I))
  200       C
           C COPY DURATION
           1011 CONTINUE
                DO 15 J=1,NTRA
                FAULTT(J,I,2)=FAULT(J)
  205           15 CONTINUE
                 2 CONTINUE
           C
           C SO AT HIS POINT, SPORTSFANS, HE HAVE CREATED THO SEPERATE FAULT       SP03APR4
           C TABLES, ONE FOR PERMANENTS AND ONE FOR TRANSIENTS -- NOW IS THE EVER  SP03APR4
  210      C PRESENT TIME TO MAKE THINGS PERFECTLY CLEAR BY MERGING THE THO TABLES SP03APR4
           C INTO ONE GRANDIOUS TIME-ORDERED FAULT TABLE.                          SP03APR4
                DO 21 I=1,NMODU
                MERG1(I)=1
                MERG2(I)=1
  215           21 CONTINUE
                DO 29 J=1,NFAUT
                RMINP=RTIME
                RMINT=RTIME
           C THE FOLLOWING LOOPS FINDS THE JTH FAULT
  220           DO 22 I=1,NMODU
```

```
                IMERG1=MERG1(I)
                IF(FAULTP(IMERG1,I).GT.RMINP) GO TO 23
                RMINP=FAULTP(IMERG1,I)
                MP=I
225           23 CONTINUE
                IMERG2=MERG2(I)
                IF(FAULTT(IMERG2,I,1).GT.RMINT) GOTO 22
                RMINT=FAULTT(IMERG2,I,1)
                MT=I
230           22 CONTINUE
                TABLE(J,1)=AMIN1(RMINP,RMINT)
              C IF NO MORE FAULT RETURN
                IF ((TABLE(J,1).GE.RTIME).AND.(J.LT.NFAUT)) RETURN
                IF (J.LT.NFAUT) GO TO 28
235             IFULL=1
                RETURN
              28 CONTINUE
                IF (RMINP.GE.RMINT)GO TO 25
                IF(MP.EQ.4) CALL PFISO(IUNIT,1)
240             IF(MP.NE.4) IUNIT=IRAN(1,NA(MP))
                TABLE(J,4)=IUNIT
                TABLE(J,2)=36.E12
                TABLE(J,3)=MP
                MERG1(MP)=MERG1(MP)+1
245             GO TO 26
              25 CONTINUE
                IMERG2=MERG2(MT)
                TABLE(J,2)=FAULTT(IMERG2,MT,2)
                TABLE(J,3)=MT
250             TABLE(J,4)=FAULTT(IMERG2,MT,3)
                MERG2(MT)=MERG2(MT)+1
              26 CONTINUE
              29 CONTINUE
                RETURN
255           C                    **********************************                  SP02APR4
              C                    *          F O R M A T S         *                  SP02APR4
              C                    **********************************                  SP02APR4
                3 FORMAT (3I1)
               19 FORMAT(2E10.3)
260            105 FORMAT (E10.3)
               949 FORMAT(17X,18HTRANSIENT DURATION,3X,E9.2,
                  1 27H MILLISECONDS (EXPONENTIAL))
               950 FORMAT(28H AVERAGE TRANSIENT DURATION ,F7.0,
                  1 23H MILLISECONDS (UNIFORM))
265            980 FORMAT(17X,14HTRANSIENT RATE,7X,E9.2,9H PER HOUR)
               990 FORMAT(5X,6HMODULE,I3,17H : PERMANENT RATE,7X,E9.2,9H PER HOUR)
               997 FORMAT(17X,12HDORMANT RATE,9X,E9.2,9H PER HOUR)
               998 FORMAT(17X,21HBURST OCCURRENCE RATE,E9.2,9H PER HOUR/
                  1 17X,14HBURST DURATION,3X,F8.1,22H SECONDS (EXPONENTIAL)/
270               2 17X,16HBURST FAULT RATE,5X,E9.2,11H PER SECOND)
               999 FORMAT(3E10.3)
                END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | FAUGEN | 1 | 71 | 79 | 118 | 134 | 177 | 233 | 236 | 254 | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 24 | AVDUR | REAL | ARRAY | COI4 | REFS | 35 | 195 | 199 | DEFINED | 194 | |
| 1056 | AVER | REAL | | | REFS | 184 | DEFINED | 183 | | | |
| 1135 | BURST | REAL | ARRAY | | REFS | 43 | 148 | 162 | DEFINED | 146 | |
| 1077 | DDUR | INTEGER | ARRAY | | REFS | 40 | 103 | DEFINED | 95 | | |
| 1142 | DEBBU | REAL | ARRAY | | REFS | 44 | 151 | 154 | 2*160 | 166 | |
| 2265 | DELAY | REAL | | CON1 | REFS | 33 | | | | | |
| 1065 | DPER | INTEGER | ARRAY | | REFS | 40 | 97 | DEFINED | 95 | | |
| 1072 | DTRA | INTEGER | ARRAY | | REFS | 40 | 100 | DEFINED | 95 | | |
| 1130 | DURA | REAL | ARRAY | | REFS | 43 | 148 | 150 | DEFINED | 146 | |
| 2262 | EXTEN | REAL | | COM1 | REFS | 33 | | | | | |
| 7506 | FAULT | REAL | ARRAY | | REFS | 45 | 68 | 74 | 76 | 82 | 86 | 115 |
| | | | | | 121 | 131 | 137 | 162 | 165 | 166 | 188 | 199 |
| | | | | | 204 | | | | | | | |
| 6130 | FAULTP | REAL | ARRAY | | REFS | 45 | 222 | 223 | DEFINED | 74 | 121 | |
| 1616 | FAULTT | REAL | ARRAY | | REFS | 45 | 227 | 228 | 248 | 250 | | |
| | | | | | DEFINED | 82 | 86 | 88 | 89 | 137 | 139 | 166 |
| | | | | | 167 | 173 | 204 | | | | | |
| 1 | FFFF | REAL | | CO40 | REFS | 38 | | | | | | |
| 1042 | I | INTEGER | | | REFS | 54 | 74 | 82 | 86 | 88 | 89 | 93 |
| | | | | | 3*95 | 97 | 100 | 103 | 109 | 2*110 | 2*111 | 113 |
| | | | | | 114 | 115 | 121 | 128 | 2*129 | 130 | 131 | 137 |
| | | | | | 139 | 3*146 | 2*147 | 3*148 | 151 | 158 | 162 | 166 |
| | | | | | 167 | 173 | 2*182 | 2*183 | 2*188 | 194 | 195 | 199 |
| | | | | | 204 | 213 | 214 | 221 | 222 | 223 | 224 | 226 |
| | | | | | 227 | 228 | 229 | DEFINED | 50 | 212 | 220 | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 33 | | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 33 | | | | | | |
| 1 | IFULL | INTEGER | | COM2 | REFS | 34 | DEFINED | 70 | 78 | 117 | 133 | 176 |
| | | | | | 235 | | | | | | | |
| 0 | IIII | INTEGER | | CO40 | REFS | 38 | | | | | | |
| 1061 | IMERG1 | INTEGER | | | REFS | 222 | 223 | DEFINED | 221 | | | |
| 1063 | IMERG2 | INTEGER | | | REFS | 227 | 228 | 248 | 250 | DEFINED | 226 | 247 |
| 1047 | IUNIT | INTEGER | | | REFS | 87 | 88 | 239 | 241 | DEFINED | 240 | |
| 1046 | J | INTEGER | | | REFS | 2*74 | 2*86 | 88 | 89 | 2*121 | 2*137 | 139 |
| | | | | | 154 | 2*160 | 166 | 2*204 | 231 | 2*233 | 234 | 241 |
| | | | | | 242 | 243 | 248 | 249 | 250 | DEFINED | 73 | 85 |
| | | | | | 120 | 136 | 153 | 203 | 216 | | | |
| 1050 | K | INTEGER | | | REFS | 98 | 101 | 104 | 165 | 166 | | |
| | | | | | DEFINED | 97 | 100 | 103 | 164 | | | |
| 1052 | L | INTEGER | | | REFS | 166 | 167 | 168 | 169 | 173 | 174 | 175 |
| | | | | | DEFINED | 152 | 168 | | | | | |
| 1055 | M | INTEGER | | | REFS | 167 | DEFINED | 163 | | | | |
| 1104 | MERG1 | INTEGER | ARRAY | | REFS | 41 | 221 | 244 | DEFINED | 213 | 244 | |
| 1111 | MERG2 | INTEGER | ARRAY | | REFS | 41 | 226 | 247 | 251 | DEFINED | 214 | 251 |
| 0 | MIS | INTEGER | | COM2 | REFS | 34 | 49 | | | | | |
| 0 | MODSIM | INTEGER | | CO28 | REFS | 36 | | | | | | |
| 1062 | MP | INTEGER | | | REFS | 239 | 2*240 | 243 | 2*244 | DEFINED | 224 | |
| 1064 | MT | INTEGER | | | REFS | 247 | 248 | 249 | 250 | 2*251 | | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | DEFINED | 229 | | | | | | |
| 1116 | NA | INTEGER | ARRAY | | REFS | 42 | 240 | DEFINED | 93 | | | |
| 0 | NAC | INTEGER | | F.P. | REFS | 56 | DEFINED | 1 | | | | |
| 1045 | NACTIV | INTEGER | | | REFS | 93 | 115 | 131 | 139 | 151 | 163 | |
| | | | | | DEFINED | 56 | 59 | | | | | |
| 0 | NFAUT | INTEGER | | F.P. | REFS | 216 | 233 | 234 | DEFINED | 1 | | |
| 0 | NFCV | INTEGER | ARRAY | FLTMIS | REFS | 39 | | | | | | |
| 7 | NIO | INTEGER | | C040 | REFS | 38 | 59 | | | | | |
| 1041 | NMIS | INTEGER | | | REFS | 62 | 94 | 108 | 127 | 145 | 181 | 193 |
| | | | | | DEFINED | 49 | | | | | | |
| 0 | NMODU | INTEGER | | F.P. | REFS | 50 | 212 | 220 | DEFINED | 1 | | |
| 2 | NNNN | INTEGER | | C040 | REFS | 38 | | | | | | |
| 1043 | NPER | INTEGER | | | REFS | 68 | 69 | 73 | 115 | 116 | 120 | |
| | | | | | DEFINED | 51 | | | | | | |
| 1 | NSPA | INTEGER | | C028 | REFS | 36 | 112 | | | | | |
| 12 | NSYSF | INTEGER | ARRAY | FLTMIS | REFS | 39 | | | | | | |
| 1044 | NTRA | INTEGER | | | REFS | 76 | 77 | 81 | 85 | 131 | 132 | 136 |
| | | | | | 187 | 188 | 198 | 199 | 203 | DEFINED | 52 | 174 |
| 1054 | NTRB | INTEGER | | | REFS | 162 | 164 | DEFINED | 161 | | | |
| 1051 | NTRC | INTEGER | | | REFS | 151 | 153 | DEFINED | 150 | | | |
| 24 | PFRIO | REAL | | FLTMIS | REFS | 39 | 63 | 65 | 68 | DEFINED | 65 | |
| 2261 | PTR | REAL | | COM1 | REFS | 33 | | | | | | |
| 2264 | PECOV | REAL | | COM1 | REFS | 33 | | | | | | |
| 1123 | RLAMB | REAL | ARRAY | | REFS | 43 | 147 | 148 | 151 | DEFINED | 146 | 147 |
| 0 | PLAMDP | REAL | ARRAY | C014 | REFS | 35 | 110 | 111 | 115 | DEFINED | 109 | 110 |
| 5 | RLAMDT | REAL | ARRAY | C014 | REFS | 35 | 129 | 130 | 131 | DEFINED | 128 | 129 |
| 17 | RMAXI | REAL | ARRAY | C014 | REFS | 35 | 183 | 186 | DEFINED | 182 | | |
| 12 | RMINI | REAL | ARRAY | C014 | REFS | 35 | 183 | 186 | DEFINED | 182 | | |
| 1057 | RMINP | REAL | | | REFS | 222 | 231 | 238 | DEFINED | 217 | 223 | |
| 1060 | RMINT | REAL | | | REFS | 227 | 231 | 238 | DEFINED | 218 | 228 | |
| 0 | RMUP | REAL | ARRAY | C033 | REFS | 37 | 114 | DEFINED | 113 | | | |
| 0 | RTIME | REAL | | F.P. | REFS | 68 | 76 | 115 | 131 | 151 | 154 | 173 |
| | | | | | 217 | 218 | 233 | DEFINED | 1 | | | |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 33 | 233 | DEFINED | 231 | 241 | 242 | 243 |
| | | | | | 248 | 249 | 250 | | | | | |
| 25 | TFRIO | REAL | | FLTMIS | REFS | 39 | 63 | 66 | 76 | DEFINED | 66 | |
| 2266 | TIME | REAL | | COM1 | REFS | 33 | | | | | | |
| 1053 | U | REAL | | | REFS | 157 | 158 | 160 | 162 | 165 | | |
| | | | | | DEFINED | 156 | 158 | 160 | | | | |

| FILE NAMES | NODE | | | | | | |
|---|---|---|---|---|---|---|---|
| INPUT | FMT | READS | 95 | 109 | 113 | 128 | 146 | 182 | 194 |
| OUTPUT | FMT | WRITES | 111 | 114 | 130 | 148 | 184 | 195 | |

| EXTERNALS | TYPE | ARGS | | REFERENCES | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ALOG | REAL | 1 | LIBRARY | 158 | | | | | |
| EXOUR | | 3 | | 199 | | | | | |
| EXPON | | 5 | | 68 | 76 | 115 | 131 | 151 | 162 |
| IRAN | INTEGER | 2 | | 139 | 163 | 240 | | | |
| PFCBCF | | 0 | | 64 | | | | | |
| PFISO | | 2 | | 239 | | | | | |
| RANF | REAL | 1 | | 156 | | | | | |
| RDIOFR | | 2 | | 63 | | | | | |
| TFISO | | 2 | | 87 | | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| UNIF | | 4 | 188 |

| INLINE FUNCTIONS | TYPE | ARGS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|---|---|
| AMIN1 | REAL | 0 | INTRIN | 160 | 231 | |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 140 | 1 | | 96 | 94 | | | | | |
| 522 | 2 | | 206 | 50 | 83 | 91 | 104 | 187 | 198 |
| 740 | 3 | FMT | 258 | 95 | | | | | |
| 172 | 4 | | 107 | 98 | | | | | |
| 32 | 5 | | 55 | 3*54 | | | | | |
| 150 | 6 | | 99 | 98 | 123 | | | | |
| 250 | 7 | | 126 | 101 | | | | | |
| 34 | 8 | | 58 | 54 | | | | | |
| 161 | 9 | | 102 | 101 | 141 | 175 | | | |
| 435 | 10 | | 180 | 104 | | | | | |
| 0 | 11 | | 122 | 120 | | | | | |
| 277 | 12 | | 135 | 112 | | | | | |
| 0 | 13 | | 140 | 136 | | | | | |
| 240 | 14 | | 119 | 116 | | | | | |
| 0 | 15 | | 205 | 203 | | | | | |
| 36 | 16 | | 61 | 54 | | | | | |
| 122 | 17 | | 92 | 57 | 60 | | | | |
| 742 | 19 | FMT | 259 | 182 | | | | | |
| 0 | 21 | | 215 | 212 | | | | | |
| 564 | 22 | | 230 | 220 | 227 | | | | |
| 553 | 23 | | 225 | 222 | | | | | |
| 630 | 25 | | 246 | 238 | | | | | |
| 640 | 26 | | 252 | 245 | | | | | |
| 603 | 28 | | 237 | 234 | | | | | |
| 0 | 29 | | 253 | 216 | | | | | |
| 470 | 30 | | 192 | 104 | | | | | |
| 314 | 31 | | 144 | 101 | | | | | |
| 343 | 32 | | 149 | 145 | | | | | |
| 421 | 35 | | 171 | 153 | 165 | | | | |
| 354 | 36 | | 155 | 157 | | | | | |
| 0 | 37 | | 170 | 164 | | | | | |
| 424 | 38 | | 172 | 154 | 169 | | | | |
| 230 | 44 | | 115 | 108 | 112 | | | | |
| 267 | 77 | | 131 | 127 | | | | | |
| 744 | 105 | FMT | 260 | 109 | 113 | 128 | 194 | | |
| 746 | 949 | FMT | 261 | 195 | | | | | |
| 756 | 950 | FMT | 263 | 184 | | | | | |
| 766 | 980 | FMT | 265 | 130 | | | | | |
| 773 | 990 | FMT | 266 | 111 | | | | | |
| 1002 | 997 | FMT | 267 | 114 | | | | | |
| 1007 | 998 | FMT | 268 | 148 | | | | | |
| 1027 | 999 | FMT | 271 | 146 | | | | | |
| 460 | 1010 | | 185 | 181 | | | | | |
| 513 | 1011 | | 202 | 189 | | | | | |
| 57 | 2010 | | 72 | 69 | | | | | |
| 50 | 2016 | | 67 | 62 | | | | | |
| 0 | 2020 | | 75 | 73 | | | | | |
| 75 | 2030 | | 80 | 77 | | | | | |

| STATEMENT LABELS | | OFF LINE | REFERENCES |
|---|---|---|---|
| 103 | 2035 | 84 | 81 |
| 0 | 2040 | 90 | 85 |
| 505 | 3030 | 196 | 193 |

| LOOPS | LABEL | | INDEX | FROM-TO | LENGTH | PROPERTIES | | | |
|---|---|---|---|---|---|---|---|---|---|
| 17 | 2 | * | I | 50 206 | 5068 | | EXT REFS | EXITS | NOT INNER |
| 64 | 2020 | | J | 73 75 | 28 | INSTACK | | | |
| 104 | 2040 | * | J | 85 90 | 158 | | EXT REFS | | |
| 245 | 11 | | J | 120 122 | 28 | INSTACK | | | |
| 300 | 13 | * | J | 136 140 | 148 | | EXT REFS | | |
| 351 | 35 | * | J | 153 171 | 538 | | EXT REFS | EXITS | NOT INNER |
| 405 | 37 | * | K | 164 170 | 148 | OPT | EXITS | | |
| 520 | 15 | | J | 203 205 | 28 | INSTACK | | | |
| 530 | 21 | | I | 212 215 | 29 | INSTACK | | | |
| 533 | 29 | * | J | 216 253 | 1108 | | EXT REFS | EXITS | NOT INNER |
| 542 | 22 | * | I | 220 230 | 248 | OPT | | | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS | NAME | (LENGTH) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 | IDIM | (1) | 1 | TABLE | (1200) | 1201 | PTR | (1) | |
| | | 1202 | EXTEN | (1) | 1203 | IDETEC | (1) | 1204 | RECOV | (1) | |
| | | 1205 | DELAY | (1) | 1206 | TIME | (1) | | | | |
| COM2 | 2 | 0 | MIS | (1) | 1 | IFULL | (1) | | | | |
| CO14 | 25 | 0 | RLAMDP | (5) | 5 | RLAMDT | (5) | 10 | RHINI | (5) | |
| | | 15 | RMAXI | (5) | 20 | AVDUR | (5) | | | | |
| CO28 | 2 | 0 | MODSIM | (1) | 1 | NSPA | (1) | | | | |
| CO33 | 5 | 0 | PMUP | (5) | | | | | | | |
| CO40 | 4 | 0 | IIII | (1) | 1 | FFFF | (1) | 2 | NNNN | (1) | |
| | | 3 | NIO | (1) | | | | | | | |
| FLTMIS | 22 | 0 | NFCV | (10) | 10 | NSYSF | (10) | 20 | PFRIO | (1) | |
| | | 21 | TFRIO | (1) | | | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 77578 | 4079 |
| COMMON LENGTH | 23638 | 1267 |

```
      SUBROUTINE FCBFLT(TIME,PLACE,DUR,NEXT)
C
C THIS SUBROUTINE IS INVOKED BY FIFAU UPON THE OCCURRENCE OF A FAULT
C IN THE FLIGHT CRITAL BUS PARTITION.  THE APPROPRIATE SUBROUTINE
C IS INVOKED TO STMULATE THE SYSTEMS RESPONSE TO THE FAULT.  FCBFLT
C THEN CHECKS THE RESULTING SYSTEM STATUS.  IF A FLIGHT CRITAL FAILURE
C OCCURRED, THE APPROPRIATE COUNTER IS INCREMENTED, AND NEXT IS SET TO
C *5* TO INDICATE SYSTEM FAILURE.  CONTROL RETURNS TO FIFAU.
C
C*************************************************************************
      INTEGER REASON
      INTEGER PLACE,STS,FLTTYP,GROUP
      INTEGER FCBSF,FALFCB
      COMMON/FCBCNT/FCBSF,FALFCB(50)
      COMMON/FAULT/FLTTYP
      COMMON/DEBUG/IDEBUG
      COMMON/STATUS/STS(20)
      COMMON/COM7/REASON
      COMMON/CO36/RMISTM
      DIMENSION PLACE(5)
C
C
      GROUP=PLACE(2)
      FLTTYP=1
      IF(DUR.GE.RMISTM) FLTTYP=2
      GOTO(10,20,30,40,50,60),GROUP
C           ...BUS FAULT
   10 CONTINUE
      CALL BUSFLT(PLACE(3))
      GOTO 200
C           ...MDM FAULT
   20 CONTINUE
      CALL MDMFLT(PLACE(3),PLACE(4))
      GOTO 200
C           ...DDU FAULT
   30 CONTINUE
      CALL DDUFLT(PLACE(3),PLACE(4))
      GOTO 200
C           ...DEDICATED DEVICE FAILURE...FF-MDM
   40 CONTINUE
      CALL DFFFLT(PLACE(3),PLACE(4))
      GOTO 200
C           ...NON-DEDICATED DEVICE   FF-MDM
   50 CONTINUE
      CALL NFFFLT(PLACE(3),PLACE(4),PLACE(5))
      GOTO 200
C           ...DEDICATED DEVICE   FA-MDM
   60 CONTINUE
      CALL DFAFLT(PLACE(3),PLACE(4))
      GOTO 200
C
C
  200 CONTINUE
      IF(STS(1).EQ.0) RETURN
      REASON=3
```

```
                NEXT=5
                FCBSF=FCBSF+1
                N=STS(1)+1
                K=STS(N)
60              FALFCB(K)=FALFCB(K)+1
                RETURN
                END
```

SYMBOLIC REFERENCE MAP

```
ENTRY POINTS     DEF LINE     REFERENCES
    2  FCBFLT          1         54        61

VARIABLES     SN  TYPE           RELOCATION
    0  AUR        REAL                F.P.        REFS       25     DEFINED       1
    1  FALFCB     INTEGER    ARRAY    FCBCNT      REFS       13        14        60     DEFINED       60
    0  FCBSF      INTEGER             FCBCNT      REFS       13        14        57     DEFINED       57
    0  FLTTYP     INTEGER             FAULT       REFS       12        15     DEFINED      24        25
  104  GROUP      INTEGER                         REFS       12        26     DEFINED      23
    0  IDEBUG     INTEGER             DEBUG       REFS       16
  106  K          INTEGER                         REFS      2*60    DEFINED      59
  105  N          INTEGER                         REFS       59     DEFINED      58
    0  NEXT       INTEGER              F.P.      DEFINED      1        56
    0  PLACE      INTEGER    ARRAY     F.P.        REFS       12        20        23        29       2*33      2*37      2*41
                                                    3*45      2*49    DEFINED       1
    0  REASON     INTEGER             COM7        REFS       11        18     DEFINED      55
    0  RMISTM     REAL                C036        REFS       19        25
    0  STS        INTEGER    ARRAY    STATUS      REFS       12        17        54        58        59
    0  TIME       REAL       *UNUSED   F.P.      DEFINED      1

EXTERNALS          TYPE    ARGS       REFERENCES
        BUSFLT               1            29
        DBUFLT               2            37
        DFAFLT               2            49
        DFFFLT               2            41
        MDMFLT               2            33
        NFFFLT               3            45

STATEMENT LABELS              DEF LINE   REFERENCES
   34  10                        28        26
   37  20                        32        26
   42  30                        36        26
   45  40                        40        26
   50  50                        44        26
   53  60                        48        26
   55  200                       53        30        34        38        42        46        50

COMMON BLOCKS   LENGTH     MEMBERS - BIAS NAME(LENGTH)
        FCBCNT      51              0 FCBSF  (1)                1 FALFCB (50)
        FAULT        1              0 FLTTYP (1)
        DEBUG        1              0 IDEBUG (1)
        STATUS      20              0 STS    (20)
        COM7         1              0 REASON (1)
        C036         1              0 RMISTM (1)

STATISTICS
  PROGRAM LENGTH    1250      85
   COMMON LENGTH    113B      75
```

```
      SUBROUTINE FCBIN
      COMMON/IFC81/BUSSTS(8),FCB(8,10),BTUSTS(10)
      COMMON/FCB2/NBTU,BTUTYP(10),BTUCON(10,4),BTUNO(10)
      COMMON/FCB3/BUSTFL,BUSCOV
      COMMON/FCB5/BTUCOV(10,2),BTUTFL(10,2)
      COMMON/IFC87/NDDUDV,DDUDVS(4),DDUDV(2,4)
      COMMON/IFC88/NDFFDV,DFFDVS(6),DFFDV(4,6)
      COMMON/IFC89/NNFFDV,NFFDVS(2,4),NFFDV(3,2,4)
      COMMON/IFC810/NDFADV,DFADVS(3),DFADV(4,3)
      COMMON/FCB12/DDUCOV(4),DDUTFL(4),DDUTFD(4,2)
      COMMON/FCB14/DFFTFD(6,4),DFFTFL(6),DFFCOV(6,3)
      COMMON/FCB16/NFFTFD(4,2),NFFTFL(4,2),NFFCOV(4),NFFPFD(4,2)
      COMMON/FCB18/DFATFD(3),DFATFL(3),DFACOV(3)
      COMMON/FCBNM/DDUNM(4),DFFNM(6),NFFNM(4),DFANM(3)
      COMMON/FCBNM1/BUSNM(8),MDDUNM(2),MFFNM(4),MFANM(4)
      INTEGER BUSNM
      DIMENSION ICNT(3),IREC(6)
      INTEGER DDUNM,DFFNM,NFFNM,DFANM
      REAL NFFTFD,NFFTFL,NFFCOV,NFFPFD
      INTEGER BUSSTS,FCB,BTUSTS,BTUTYP,BTUCON,BTUNO
      INTEGER DDUDVS,DDUDV,DFFDVS,DFFDV,DFADVS,DFADV
      DATA BUSSTS/8*1/
      NTYPS=3
      READ 2001,NBTU
      DO 10 I=1,8
        READ 2002,(FCB(I,J),J=1,NBTU)
   10 CONTINUE
      READ 2002,(BTUTYP(I),I=1,NBTU)
      DO 20 I=1,NTYPS
        ICNT(I)=0
   20 CONTINUE
      DO 50 J=1,NBTU
        ITYP=BTUTYP(J)
        IF(ITYP.GT.NTYPS) STOP 1021
        ICNT(ITYP)=ICNT(ITYP)+1
        BTUNO(J)=ICNT(ITYP)
        ISUM=0
        DO 40 I=1,8
          IPRT=FCB(I,J)
          IF(IPRT.EQ.0) GOTO 40
      IF(IPRT.LT.0.OR.IPRT.GT.2) STOP 1022
          ISUM=ISUM+1
          BTUCON(J,ISUM)=I
   40   CONTINUE
        BTUSTS(J)=ISUM
   50 CONTINUE
      READ 2001,NDDUDV
      IF((NDDUDV.LT.0).OR.(NDDUDV.GT.4)) STOP 1023
      DO 60 I=1,2
        READ 2002,(DDUDV(I,J),J=1,NDDUDV)
   60  CONTINUE
      DO 80 J=1,NDDUDV
        ISUM=0
        DO 70 I=1,2
          IF(DDUDV(I,J).NE.0)ISUM=ISUM+1
```

```
 70    CONTINUE
       NDUDVS(J)=ISUM
 80 CONTINUE
       READ 2001,NOFFDV
60     IF(NOFFDV.LT.0.OR.NOFFDV.GT.6) STOP 1024
       DO 90 I=1,4
         READ 2002,(OFFDV(I,J),J=1,NOFFDV)
 90 CONTINUE
       DO 110 J=1,NOFFDV
65       ISUM=0
         DO 100 I=1,4
           IF(OFFDV(I,J).NE.0) ISUM=ISUM+1
100    CONTINUE
         OFFDVS(J)=ISUM
70  110 CONTINUE
       READ 2001,NNFFDV
       IF(NNFFDV.LT.0.OR.NNFFDV.GT.4) STOP 1025
       READ 2002,(IREC(I),I=1,NNFFDV)
       DO 130 J=1,NNFFDV
75       DO 120 I=1,2
         I1=0
         I2=0
           IF(IREC(J).LE.0)GOTO 115
             IREC(J)=IREC(J)-1
80           I1=3
             I2=1
115      CONTINUE
         NFFDVS(I,J)=I1
         DO 120 K=1,3
85         NFFDV(K,I,J)=I2
120    CONTINUE
130 CONTINUE
       READ 2001,NDFADV
       IF((NDFADV.LE.0).OR.(NDFADV.GT.3)) STOP 1026
90     DO 140 I=1,4
         READ 2002,(DFADV(I,J),J=1,NDFADV)
140 CONTINUE
       DO 150 J=1,NDFADV
         ISUM=0
95       DO 145 I=1,4
           IF(DFADV(I,J).NE.0) ISUM=ISUM+1
145    CONTINUE
         DFADVS(J)=ISUM
150 CONTINUE
100    READ 2004,(BUSNM(I),I=1,8)
       READ 2004,(NDDUNM(I),I=1,2)
       READ 2004,(NFFNM(I),I=1,4)
       READ 2004,(NFANM(I),I=1,4)
       READ 2004,(NDUNM(I),I=1,NDDUDV)
105    READ 2004,(OFFNM(I),I=1,NOFFDV)
       READ 2004,(NFFNM(I),I=1,NNFFDV)
       READ 2004,(DFANM(I),I=1,NDFADV)
       READ 2003,BUSTFL,BUSCOV
       DO 160 J=1,NBTU
110      READ 2003,BTUTFL(J,1),BTUTFL(J,2),BTUCOV(J,1),BTUCOV(J,2)
```

```
      160 CONTINUE
          READ 2003,(DDUTFL(I),I=1,NDDUDV)
          READ 2003,(DDUCOV(I),I=1,NDDUDV)
          READ 2003,(DDUTFD(I,1),I=1,NDDUDV)
          READ 2003,(DDUTFD(I,2),I=1,NDDUDV)
          READ 2003,(DFFTFL(I),I=1,NDFFDV)
          DO 170 J=1,3
            READ 2003,(DFFCOV(I,J),I=1,NDFFDV)
      170 CONTINUE
          DO 180 J=1,4
            READ 2003,(DFFTFD(I,J),I=1,NDFFDV)
      180 CONTINUE
          READ 2003,(NFFTFL(I,1),I=1,NNFFDV)
          READ 2003,(NFFTFL(I,2),I=1,NNFFDV)
          READ 2003,(NFFTFD(I,1),I=1,NNFFDV)
          READ 2003,(NFFTFD(I,2),I=1,NNFFDV)
          READ 2003,(NFFPFD(I,1),I=1,NNFFDV)
          READ 2003,(NFFCOV(I),I=1,NNFFDV)
          READ 2003,(DFATFL(I),I=1,NDFADV)
          READ 2003,(DFATFD(I),I=1,NDFADV)
          READ 2003,(DFACOV(I),I=1,NDFADV)
          RETURN
     2001 FORMAT(8I10)
     2002 FORMAT(20I2)
     2003 FORMAT(8F10.0)
     2004 FORMAT(8A10)
          END
```

SYMBOLIC REFFRENCE MAP

ENTRY POINTS DEF LINE REFERENCES
1 FCBIN 1 132

VARIABLES SN TYPE RELOCATION

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | BTUCON | INTEGER | ARRAY | FCB2 | REFS | 3 | 20 | DEFINED | 43 | | | |
| 0 | BTUCOV | REAL | ARPAY | FCB5 | REFS | 5 | DEFINED | 2*110 | | | | |
| 63 | BTUNO | INTEGER | ARRAY | FCB2 | REFS | 3 | 20 | DEFINED | 36 | | | |
| 130 | BTUSTS | INTEGER | ARRAY | IFCB1 | REFS | 2 | 20 | DEFINED | 45 | | | |
| 24 | BTUTFL | REAL | ARRAY | FCB5 | REFS | 5 | DEFINED | 2*110 | | | | |
| 1 | BTUTYP | INTEGER | ARRAY | FCB2 | REFS | 3 | 20 | 33 | DEFINED | 28 | | |
| 1 | BUSCOV | REAL | | FCB3 | REFS | 4 | DEFINED | 108 | | | | |
| 0 | BUSNM | INTEGER | ARRAY | FCBNM1 | REFS | 15 | 16 | DEFINED | 100 | | | |
| 0 | BUSSTS | INTEGER | ARRAY | IFCB1 | REFS | 2 | 20 | DEFINED | 22 | | | |
| 0 | BUSTFL | REAL | | FCB3 | REFS | 4 | DEFINED | 108 | | | | |
| 0 | DDUCOV | REAL | ARRAY | FCB12 | REFS | 10 | DEFINED | 113 | | | | |
| 5 | DDUDV | INTEGER | ARRAY | IFCB7 | REFS | 6 | 21 | 55 | DEFINED | 50 | | |
| 1 | DDUDVS | INTEGER | ARRAY | IFCB7 | REFS | 6 | 21 | DEFINED | 57 | | | |
| 0 | DDUNM | INTEGER | ARRAY | FCBNM | REFS | 14 | 18 | DEFINED | 104 | | | |
| 10 | DDUTFD | REAL | ARRAY | FCB12 | REFS | 10 | DEFINED | 114 | 115 | | | |
| 4 | DDUTFL | REAL | ARRAY | FCB12 | REFS | 10 | DEFINED | 112 | | | | |
| 6 | DFACOV | REAL | ARRAY | FCB10 | REFS | 13 | DEFINED | 131 | | | | |
| 4 | DFADV | INTEGER | ARRAY | IFCB10 | REFS | 9 | 21 | 96 | DEFINED | 91 | | |
| 1 | DFADVS | INTEGER | ARRAY | IFCB10 | REFS | 9 | 21 | DEFINED | 98 | | | |
| 16 | DFANM | INTEGER | ARRAY | FCBNM | REFS | 14 | 18 | DEFINED | 107 | | | |
| 0 | DFATFD | REAL | ARRAY | FCB10 | REFS | 13 | DEFINED | 130 | | | | |
| 3 | DFATFL | REAL | ARRAY | FCB10 | REFS | 13 | DEFINED | 129 | | | | |
| 36 | DFFCOV | REAL | ARRAY | FCB14 | REFS | 11 | DEFINED | 118 | | | | |
| 7 | DFFDV | INTEGER | ARRAY | IFCB8 | REFS | 7 | 21 | 67 | DEFINED | 62 | | |
| 1 | DFFDVS | INTEGER | ARRAY | IFCB8 | REFS | 7 | 21 | DEFINED | 69 | | | |
| 4 | DFFNM | INTEGER | ARRAY | FCBNM | REFS | 14 | 18 | DEFINED | 105 | | | |
| 0 | DFFTFD | REAL | ARRAY | FCB14 | REFS | 11 | DEFINED | 121 | | | | |
| 30 | DFFTFL | REAL | ARRAY | FCB14 | REFS | 11 | DEFINED | 116 | | | | |
| 10 | FCB | INTEGER | ARRAY | IFCB1 | REFS | 2 | 20 | 39 | DEFINED | 26 | | |
| 572 | I | INTEGER | | | REFS | 26 | 28 | 30 | 39 | 43 | 50 | 55 |
| | | | | | 62 | 67 | 73 | 83 | 85 | 91 | 96 | 100 |
| | | | | | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 112 |
| | | | | | 113 | 114 | 115 | 116 | 118 | 121 | 123 | 124 |
| | | | | | 125 | 126 | 127 | 128 | 129 | 130 | 131 | |
| | | | | | DEFINED | 25 | 28 | 29 | 38 | 49 | 54 | 61 |
| | | | | | 66 | 73 | 75 | 90 | 95 | 100 | 101 | 102 |
| | | | | | 103 | 104 | 105 | 106 | 107 | 112 | 113 | 114 |
| | | | | | 115 | 116 | 118 | 121 | 123 | 124 | 125 | 126 |
| | | | | | 127 | 128 | 129 | 130 | 131 | | | |
| 602 | ICNT | INTEGER | ARRAY | | REFS | 17 | 35 | 36 | DEFINED | 30 | 35 | |
| 576 | IPRT | INTEGER | | | REFS | 40 | 2*41 | DEFINED | 39 | | | |
| 605 | IREC | INTEGER | ARRAY | | REFS | 17 | 78 | 79 | DEFINED | 73 | 79 | |
| 575 | ISUM | INTEGER | | | REFS | 42 | 43 | 45 | 55 | 57 | 67 | 69 |
| | | | | | 96 | 98 | DEFINED | 37 | 42 | 53 | 55 | 65 |
| | | | | | 67 | 94 | 96 | | | | | |
| 574 | ITYP | INTEGER | | | REFS | 34 | 2*35 | 36 | DEFINED | 33 | | |
| 577 | I1 | INTEGER | | | REFS | 83 | DEFINED | 76 | 80 | | | |
| 600 | I2 | INTEGER | | | REFS | 85 | DEFINED | 77 | 81 | | | |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 573 | J | INTEGER | | | REFS | 26 | 33 | 36 | 39 | 43 | 45 | 50 |
| | | | | | | 55 | 57 | 62 | 67 | 69 | 78 | 2*79 | 83 |
| | | | | | | 85 | 91 | 96 | 98 | 4*110 | 118 | 121 | |
| | | | | | DEFINED | 26 | 32 | 50 | 52 | 62 | 64 | 74 | |
| | | | | | | 91 | 93 | 109 | 117 | 120 | | | |
| 601 | K | INTEGER | | | REFS | 85 | DEFINED | 84 | | | | | |
| 10 | MDDUNM | INTEGER | ARRAY | FCBNM1 | REFS | 15 | DEFINED | 101 | | | | | |
| 16 | MFANN | INTEGER | ARRAY | FCBNM1 | REFS | 15 | DEFINED | 103 | | | | | |
| 12 | MFFNM | INTEGER | ARRAY | FCBNM1 | REFS | 15 | DEFINED | 102 | | | | | |
| 0 | NBTU | INTEGER | | FCB2 | REFS | 3 | 26 | 28 | 32 | 109 | | | |
| | | | | | DEFINED | 24 | | | | | | | |
| 0 | NDDUOV | INTEGER | | IFCB7 | REFS | 6 | 2*48 | 50 | 52 | 104 | 112 | 113 | |
| | | | | | | 114 | 115 | DEFINED | 47 | | | | |
| 0 | NDFADV | INTEGER | | IFCB10 | REFS | 9 | 2*89 | 91 | 93 | 107 | 129 | 130 | |
| | | | | | | 131 | DEFINED | 88 | | | | | |
| 0 | NDFFDV | INTEGER | | IFCB8 | REFS | 7 | 2*60 | 62 | 64 | 105 | 116 | 118 | |
| | | | | | | 121 | DEFINED | 59 | | | | | |
| 20 | NFFCOV | REAL | ARRAY | FCB16 | REFS | 12 | 19 | DEFINED | 128 | | | | |
| 11 | NFFDV | INTEGER | ARRAY | IFCB9 | REFS | 8 | DEFINED | 85 | | | | | |
| 1 | NFFDVS | INTEGER | ARRAY | IFCB9 | REFS | 8 | DEFINED | 83 | | | | | |
| 12 | NFFNM | INTEGER | ARRAY | FCBNM | REFS | 14 | 18 | DEFINED | 106 | | | | |
| 24 | NFFPFD | REAL | ARRAY | FCB16 | REFS | 12 | 19 | DEFINED | 127 | | | | |
| 0 | NFFTFD | REAL | ARRAY | FCB16 | REFS | 12 | 19 | DEFINED | 125 | 126 | | | |
| 10 | NFFTFL | REAL | ARRAY | FCB16 | REFS | 12 | 19 | DEFINED | 123 | 124 | | | |
| 0 | NNFFDV | INTEGER | | IFCB9 | REFS | 8 | 2*72 | 73 | 74 | 106 | 123 | 124 | |
| | | | | | | 125 | 126 | 127 | 128 | DEFINED | 71 | | |
| 571 | NTYPS | INTEGER | | | REFS | 29 | 34 | DEFINED | 23 | | | | |

| FILE NAMES | MODE | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| INPUT | FMT | READS | 24 | 26 | 28 | 47 | 50 | 59 | 62 | 71 |
| | | | 73 | 88 | 91 | 100 | 101 | 102 | 103 | 104 | 105 |
| | | | 106 | 107 | 108 | 110 | 112 | 113 | 114 | 115 | 116 |
| | | | 118 | 121 | 123 | 124 | 125 | 126 | 127 | 128 | 129 |
| | | | 130 | 131 | | | | | | | |

| STATEMENT LABELS | | DEF LINE | REFEPENCES | |
|---|---|---|---|---|
| 0 | 10 | 27 | 25 | |
| 0 | 20 | 31 | 29 | |
| 72 | 40 | 44 | 38 | 40 |
| 0 | 50 | 46 | 32 | |
| 0 | 60 | 51 | 49 | |
| 0 | 70 | 56 | 54 | |
| 0 | 80 | 58 | 52 | |
| 0 | 90 | 63 | 61 | |
| 0 | 100 | 68 | 66 | |
| 0 | 110 | 70 | 64 | |
| 244 | 115 | 82 | 78 | |
| 0 | 120 | 86 | 75 | 84 |
| 0 | 130 | 87 | 74 | |
| 0 | 140 | 92 | 90 | |
| 0 | 145 | 97 | 95 | |
| 0 | 150 | 99 | 93 | |
| 0 | 160 | 111 | 109 | |
| 0 | 170 | 119 | 117 | |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 180 | | 122 | 120 | | | | | | | | |
| 561 | 2001 | FMT | 133 | 24 | 47 | 59 | 71 | 88 | | | | |
| 563 | 2002 | FMT | 134 | 26 | 28 | 50 | 62 | 73 | 91 | | | |
| 565 | 2003 | FMT | 135 | 108 | 110 | 112 | 113 | 114 | 115 | 116 | 118 | 121 |
| | | | | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 |
| 567 | 2004 | FMT | 136 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | | |
|---|---|---|---|---|---|---|---|
| 11 | 10 | * I | 25 27 | 158 | | EXT REFS | NOT INNER |
| 14 | | * J | 26 | 68 | | EXT REFS | |
| 35 | 20 | I | 29 31 | 28 | INSTACK | | |
| 40 | 50 | * J | 32 46 | 418 | | EXT REFS | NOT INNER |
| 53 | 40 | * I | 38 44 | 228 | | EXT REFS | |
| 114 | 60 | * I | 49 51 | 158 | | EXT REFS | NOT INNER |
| 117 | | * J | 50 | 68 | | EXT REFS | |
| 132 | 80 | * J | 52 58 | 148 | | NOT INNER | |
| 136 | 70 | I | 54 56 | 48 | INSTACK | | |
| 161 | 90 | * I | 61 63 | 158 | | EXT REFS | NOT INNER |
| 164 | | * J | 62 | 68 | | EXT REFS | |
| 177 | 110 | * J | 64 70 | 148 | | NOT INNER | |
| 203 | 100 | I | 66 68 | 48 | INSTACK | | |
| 233 | 130 | * J | 74 87 | 278 | | NOT INNER | |
| 234 | 120 | * I | 75 86 | 238 | | NOT INNER | |
| 253 | 120 | K | 84 86 | 28 | INSTACK | | |
| 276 | 140 | * I | 90 92 | 158 | | EXT REFS | NOT INNER |
| 301 | | * J | 91 | 68 | | EXT REFS | |
| 314 | 150 | * J | 93 99 | 148 | | NOT INNER | |
| 320 | 145 | I | 95 97 | 48 | INSTACK | | |
| 410 | 160 | * J | 109 111 | 168 | | EXT REFS | |
| 460 | 170 | * J | 117 119 | 118 | | EXT REFS | |
| 472 | 180 | * J | 120 122 | 118 | | EXT REFS | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| IFCB1 | 98 | 0 BUSSTS (8) | 0 FCB (80) | 88 BTUSTS (10) | | | |
| FCB2 | 61 | 0 NBTU (1) | 1 BTUTYP (10) | 11 BTUCON (40) | | | |
| | | 51 BTUNO (10) | | | | | |
| FCB3 | 2 | 0 BUSTFL (1) | 1 BUSCOV (1) | | | | |
| FCB5 | 40 | 0 BTUCOV (20) | 20 BTUTFL (20) | | | | |
| IFCB7 | 13 | 0 NDDUDV (1) | 1 DDUDVS (4) | 5 DDUDV (8) | | | |
| IFCB8 | 31 | 0 NDFFDV (1) | 1 DFFDVS (6) | 7 DFFDV (24) | | | |
| IFCB9 | 33 | 0 NNFFDV (1) | 1 NFFDVS (8) | 9 NFFDV (24) | | | |
| IFCB10 | 16 | 0 NDFADV (1) | 1 DFADVS (3) | 4 DFADV (12) | | | |
| FCB12 | 16 | 0 *DDUCOV* (4) | 4 *DDUTFL* (4) | 8 DDUTFD (8) | | | |
| FCB14 | 48 | 0 DFFTFD (24) | 24 DFFTFL (6) | 30 DFFCOV (18) | | | |
| FCB16 | 28 | 0 NFFTFD (8) | 8 NFFTFL (8) | 16 NFFCOV (4) | | | |
| | | 20 NFFPFD (8) | | | | | |
| FCB18 | 9 | 0 DFATFD (3) | 3 DFATFL (3) | 6 DFACOV (3) | | | |
| FCBNM | 17 | 0 DDUNM (4) | 4 DFFNM (6) | 10 NFFNM (4) | | | |
| | | 14 DFANM (3) | | | | | |
| FCBNM1 | 18 | 0 DUSNM (8) | 8 MDDUNM (2) | 10 MFFNM (4) | | | |
| | | 14 MFANM (4) | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 6278 | 407 |
| COMMON LENGTH | 6568 | 430 |

```
      SUBROUTINE FCBINI
      COMMON/FCB1/FCB1A(98)/IFCB1/FCB1B(98)
      COMMON/FCB7/FCB7A(17)/IFCB7/FCB7B(13)
      COMMON/FCB8/FCB8A(37)/IFCB8/FCB8B(31)
 5    COMMON/FCB9/FCB9A(37)/IFCB9/FCB9B(33)
      COMMON/FCB10/FCB10A(19)/IFCB10/FCB10B(16)
      COMMON/FLAGS/IFLG1,IFLG2,IFLG3,IFLG4,IFLG5
      COMMON/STATUS/STS(20)
      INTEGER FCB1A,FCB1B,FCB7A,FCB7B,FCB8A,FCB8B
10    INTEGER FCB9A,FCB9B,FCB10A,FCB10B
      LOGICAL IFLG1,IFLG2,IFLG3,IFLG4,IFLG5
      IF(IFLG1) CALL INCOPY(98,FCB1A,FCB1B)
      IF(IFLG2) CALL INCOPY(13,FCB7A,FCB7B)
      IF(IFLG3) CALL INCOPY(31,FCB8A,FCB8B)
15    IF(IFLG4) CALL INCOPY(33,FCB9A,FCB9B)
      IF(IFLG5) CALL INCOPY(16,FCB10A,FCB10B)
      IFLG1 = .FALSE.
      IFLG2 = .FALSE.
      IFLG3 = .FALSE.
20    IFLG4 = .FALSE.
      IFLG5 = .FALSE.
      STS(1)=0
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

ENTRY POINTS DEF LINE REFERENCES
1 FCBINI 1 23

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FCB1A | 0 | INTEGER | ARRAY | FCB1 | REFS | 2 | 9 | 12 | | | |
| FCB1B | 0 | INTEGER | ARRAY | IFCB1 | REFS | 2 | 9 | 12 | | | |
| FCB10A | 0 | INTEGER | ARRAY | FCB10 | REFS | 6 | 10 | 16 | | | |
| FCB10B | 0 | INTEGER | ARRAY | IFCB10 | REFS | 6 | 10 | 16 | | | |
| FCB7A | 0 | INTEGER | ARRAY | FCB7 | REFS | 3 | 9 | 13 | | | |
| FCB7B | 0 | INTEGER | ARRAY | IFCB7 | REFS | 3 | 9 | 13 | | | |
| FCB8A | 0 | INTEGER | ARRAY | FCB8 | REFS | 4 | 9 | 14 | | | |
| FCB8B | 0 | INTEGER | ARRAY | IFCB8 | REFS | 4 | 9 | 14 | | | |
| FCB9A | 0 | INTEGER | ARRAY | FCB9 | REFS | 5 | 10 | 15 | | | |
| FCB9B | 0 | INTEGER | ARRAY | IFCB9 | REFS | 5 | 10 | 15 | | | |
| IFLG1 | 0 | LOGICAL | | FLAGS | REFS | 7 | 11 | 12 | DEFINED | 17 | |
| IFLG2 | 1 | LOGICAL | | FLAGS | REFS | 7 | 11 | 13 | DEFINED | 18 | |
| IFLG3 | 2 | LOGICAL | | FLAGS | REFS | 7 | 11 | 14 | DEFINED | 19 | |
| IFLG4 | 3 | LOGICAL | | FLAGS | REFS | 7 | 11 | 15 | DEFINED | 20 | |
| IFLG5 | 4 | LOGICAL | | FLAGS | REFS | 7 | 11 | 16 | DEFINED | 21 | |
| STS | 0 | REAL | ARRAY | STATUS | REFS | 8 | DEFINED | 22 | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES | | | | |
|---|---|---|---|---|---|---|---|
| INCOPY | | 3 | 12 | 13 | 14 | 15 | 16 |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | |
|---|---|---|---|---|---|
| FCB1 | 98 | 0 FCB1A | (98) | | |
| IFCB1 | 98 | 0 FCB1B | (98) | | |
| FCB7 | 17 | 0 FCB7A | (17) | | |
| IFCB7 | 13 | 0 FCB7B | (13) | | |
| FCB8 | 37 | 0 FCB8A | (37) | | |
| IFCB8 | 31 | 0 FCB8B | (31) | | |
| FCB9 | 37 | 0 FCB9A | (37) | | |
| IFCB9 | 33 | 0 FCB9B | (33) | | |
| FCB10 | 19 | 0 FCB10A | (19) | | |
| IFCB10 | 16 | 0 FCB10B | (16) | | |
| FLAGS | 5 | 0 IFLG1 | (1) | 1 IFLG2 | (1) 2 IFLG3 (1) |
| | | 3 IFLG4 | (1) | 4 IFLG5 | (1) |
| STATUS | 20 | 0 STS | (20) | | |

STATISTICS
PROGRAM LENGTH 568 46
COMMON LENGTH 650B 424

```
      SUBROUTINE FCBPI
      COMMON/FCB4/ICNT4(3)
      COMMON/FCB6/ICNT6(60)
      COMMON/FCB11/ICNT11(12)
      COMMON/FCB13/ICNT13(18)
      COMMON/FCB15/ICNT15(24)
      COMMON/FCB17/ICNT17(9)
      COMMON/FCBCNT/IFCBCN(51)
      COMMON/FLAGS/IFLG(5)
      LOGICAL IFLG
      DO 10 I=1,5
        IFLG(I)=.TRUE.
   10 CONTINUE
      CALL FCBINI
      CALL CLEAR(3,ICNT4)
      CALL CLEAR(60,ICNT6)
      CALL CLEAR(12,ICNT11)
      CALL CLEAR(18,ICNT13)
      CALL CLEAR(24,ICNT15)
      CALL CLEAR(9,ICNT17)
      CALL CLEAR(51,IFCBCN)
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

```
ENTRY POINTS      DEF LINE     REFERENCES
    1  FCBPI           1        22

VARIABLES     SN  TYPE            RELOCATION
   64  I         INTEGER                          REFS        12    DEFINED       11
    0  ICNT11    INTEGER    ARRAY    FCB11        REFS         4         17
    0  ICNT13    INTEGER    ARRAY    FCB13        REFS         5         18
    0  ICNT15    INTEGER    ARRAY    FCB15        REFS         6         19
    0  ICNT17    INTEGER    ARRAY    FCB17        REFS         7         20
    0  ICNT4     INTEGER    ARRAY    FCB4         REFS         2         15
    0  ICNT6     INTEGER    ARRAY    FCB6         REFS         3         16
    0  IFCBCN    INTEGER    ARRAY    FCBCNT       REFS         8         21
    0  IFLG      LOGICAL    ARRAY    FLAGS        REFS         9         10    DEFINED       12

EXTERNALS,          TYPE   ARGS       REFERENCES
       CLEAR                 2            15        16        17        18        19        20        21
       FCBINI                0            14

STATEMENT LABELS              DEF LINE   REFERENCES
    0  10                        13         11

LOOPS  LABEL    INDEX     FROM-TO    LENGTH     PROPERTIES
    4  10       I           11 13        28      INSTACK

COMMON BLOCKS   LENGTH     MEMBERS - BIAS NAME(LENGTH)
       FCB4          3               0 ICNT4   (3)
       FCB6         60               0 ICNT6   (60)
       FCB11        12               0 ICNT11  (12)
       FCB13        18               0 ICNT13  (18)
       FCB15        24               0 ICNT15  (24)
       FCB17         9               0 ICNT17  (9)
       FCBCNT       51               0 IFCBCN  (51)
       FLAGS         5               0 IFLG    (5)

STATISTICS
  PROGRAM LENGTH     65B      53
   COMMON LENGTH    266B     182
```

```
      SUBROUTINE FCBPM1
      COMMON/FCB7/NDDUDV,DDUDVS(4),DDUDV(2,4),DDUDVN(4)
      COMMON/FCB8/NDFFDV,DFFDVS(6),DFFDV(4,6),DFFDVN(6)
      COMMON/FCB9/NNFFDV,NFFDVS(2,4),NFFDV(3,2,4),NFFDVN(4)
      COMMON/FCB10/NDFADV,DFADVS(3),DFADV(4,3),DFADVN(3)
      COMMON/FCBUC/FCBUCF(6)
      COMMON/CO36/RMISTM
      COMMON/FLTHIS/NFCV(10),NSYSF(10),PFRIO,TFRIO
      DIMENSION NAMEF(2)
      DIMENSION IPLACE(5)
      DATA TIME/10000.0/
      DATA RMISTM/100000.0/
      DATA DFADVN/21,22,23/
      DATA NFFDVN/17,18,19,20/
      DATA DFFDVN/11,12,13,14,15,16/
      DATA DDUDVN/7,8,9,10/
      DATA NAMEF/10HTRANSIENT ,10HPERMANENT /
      DATA FCBUCF/1,2,3,4,5,6/
      CALL FCBIN
      CALL RDIOFR(PFRIO,TFRIO)
      CALL PFCBCF
    5 CONTINUE
      READ 3001,NOMIS,NOFLT,IDEBUG
      IF(NOMIS.LE.0) RETURN
      CALL FCBPI
      DO 10 I=1,10
         NFCV(I)=0
         NSYSF(I)=0
   10 CONTINUE
      DO 100 K=1,NOMIS
         NEXT=0
         DUR=RMISTM
         CALL FCBINI
         DO 50 J=1,NOFLT
      CALL PFISO(IP,2)
      CALL UNPACK(IP,IPLACE)
            CALL FCBFLT(TIME,IPLACE,DUR,NEXT)
         NFCV(J)=NFCV(J)+1
            IF(NEXT.EQ.5) GOTO 60
   50    CONTINUE
         GO TO 100
   60    CONTINUE
         NSYSF(J)=NSYSF(J)+1
  100 CONTINUE
      CALL PIOSTS
      GOTO 5
 2000 FORMAT(1H1,10X,10HMISSION - ,I3/11X,12H-----------/)
 2001 FORMAT(1X,F10.2,3H...,A10,4X,4I3/)
 2002 FORMAT(/16H  SYSTEM FAILURE/)
 3000 FORMAT(I1,I9,F10.0)
 3001 FORMAT(3I10)
      END
```

SYMBOLIC REFERENCE MAP

ENTRY POINTS DEF LINE REFERENCES
1 FCBPM1 1 24

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | DDUDV | REAL | ARRAY | FCB7 | REFS | 2 | | | | | |
| 15 | DDUDVN | REAL | ARRAY | FCB7 | REFS | 2 | DEFINED | 16 | | | |
| 1 | DDUDVS | REAL | ARRAY | FCB7 | REFS | 2 | | | | | |
| 4 | DFADV | REAL | ARRAY | FCB10 | REFS | 5 | | | | | |
| 20 | DFADVN | REAL | ARRAY | FCB10 | REFS | 5 | DEFINED | 13 | | | |
| 1 | DFADVS | REAL | ARRAY | FCB10 | REFS | 5 | | | | | |
| 7 | DFFDV | REAL | ARRAY | FCB8 | REFS | 3 | | | | | |
| 37 | DFFDVN | REAL | ARRAY | FCB8 | REFS | 3 | DEFINED | 15 | | | |
| 1 | DFFDVS | REAL | ARRAY | FCB8 | REFS | 3 | | | | | |
| 137 | DUR | REAL | | | REFS | 37 | DEFINED | 32 | | | |
| 0 | FCBUCF | REAL | ARRAY | FCBUC | REFS | 6 | DEFINED | 18 | | | |
| 134 | I | INTEGER | | | REFS | 27 | 28 | DEFINED | 26 | | |
| 133 | IDEBUG | * INTEGER | | | DEFINED | 23 | | | | | |
| 141 | IP | INTEGER | | | REFS | 35 | 36 | | | | |
| 144 | IPLACE | INTEGER | ARRAY | | REFS | 10 | 36 | 37 | | | |
| 140 | J | INTEGER | | | REFS | 2*38 | 2*43 | DEFINED | 34 | | |
| 135 | K | * INTEGER | | | DEFINED | 30 | | | | | |
| 142 | NAMEF | INTEGER | ARRAY | | REFS | 9 | DEFINED | 17 | | | |
| 0 | NDDUDV | INTEGER | | FCB7 | REFS | 2 | | | | | |
| 0 | NDFADV | INTEGER | | FCB10 | REFS | 5 | | | | | |
| 0 | NDFFDV | INTEGER | | FCB8 | REFS | 3 | | | | | |
| 136 | NEXT | INTEGER | | | REFS | 37 | 39 | DEFINED | 31 | | |
| 0 | NFCV | INTEGER | ARRAY | FLTMIS | REFS | 8 | 38 | DEFINED | 27 | 38 | |
| 11 | NFFDV | INTEGER | ARRAY | FCB9 | REFS | 4 | | | | | |
| 41 | NFFDVN | INTEGER | ARRAY | FCB9 | REFS | 4 | DEFINED | 14 | | | |
| 1 | NFFDVS | INTEGER | ARRAY | FCB9 | REFS | 4 | | | | | |
| 0 | NNFFDV | INTEGER | | FCB9 | REFS | 4 | | | | | |
| 132 | NOFLT | INTEGER | | | REFS | 34 | DEFINED | 23 | | | |
| 131 | NOMIS | INTEGER | | | REFS | 24 | 30 | DEFINED | 23 | | |
| 12 | NSYSF | INTEGER | ARRAY | FLTMIS | REFS | 8 | 43 | DEFINED | 28 | 43 | |
| 24 | PFRIO | REAL | | FLTMIS | REFS | 8 | 20 | | | | |
| 0 | RMISTM | REAL | | C036 | REFS | 7 | 32 | DEFINED | 12 | | |
| 25 | TFRIO | REAL | | FLTMIS | REFS | 8 | 20 | | | | |
| 103 | TIME | REAL | | | REFS | 37 | DEFINED | 11 | | | |

FILE NAMES MODE
INPUT FMT READS 23

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| FCBFLT | | 4 | 37 |
| FCBIN | | 0 | 19 |
| FCBINI | | 0 | 33 |
| FCBPI | | 0 | 25 |
| PFCBCF | | 0 | 21 |
| PFISO | | 2 | 35 |
| PIOSTS | | 0 | 45 |
| RDIOFR | | 2 | 20 |
| UNPACK | | 2 | 36 |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | |
|---|---|---|---|---|---|
| 10 | 5 | | 22 | 46 | |
| 0 | 10 | | 29 | 26 | |
| 0 | 50 | | 40 | 34 | |
| 56 | 60 | | 42 | 39 | |
| 60 | 100 | | 44 | 30 | 41 |
| 104 | 2000 | FMT NO REFS | 47 | | |
| 112 | 2001 | FMT NO REFS | 48 | | |
| 116 | 2002 | FMT NO REFS | 49 | | |
| 122 | 3000 | FMT NO REFS | 50 | | |
| 125 | 3001 | FMT | 51 | 23 | |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | |
|---|---|---|---|---|---|---|
| 30 | 10 | I | 26 29 | 28 | INSTACK | |
| 33 | 100 | * K | 30 44 | 309 | EXT REFS | NOT INNER |
| 40 | 50 | * J | 34 40 | 158 | EXT REFS | EXITS |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| FCB7 | 17 | 0 | NDDUDV | (1) | 1 | DDUDVS | (4) | 5 | DDUDV (8) |
| | | 13 | DDUDVN | (4) | | | | | |
| FCB8 | 37 | 0 | NDFFDV | (1) | 1 | DFFDVS | (6) | 7 | DFFDV (24) |
| | | 31 | DFFDVN | (6) | | | | | |
| FCB9 | 37 | 0 | NNFFDV | (1) | 1 | NFFDVS | (8) | 9 | NFFDV (24) |
| | | 33 | NFFDVN | (4) | | | | | |
| FCB10 | 19 | 0 | NDFADV | (1) | 1 | DFADVS | (3) | 4 | DFADV (12) |
| | | 16 | DFADVN | (3) | | | | | |
| FCBUC | 6 | 0 | FCBUCF | (6) | | | | | |
| C036 | 1 | 0 | RMISTM | (1) | | | | | |
| FLTHIS | 22 | 0 | NFCV | (10) | 10 | NSYSF | (10) | 20 | PFRIO (1) |
| | | 21 | TFRIO | (1) | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 1518 | 105 |
| COMMON LENGTH | 213B | 139 |

```
      SUBROUTINE FIFAU(IN,NEXT,ISYNC)
C                     THIS VERSION:  MARCH 1976
C THIS SUBROUTINE LOOKS IF A FAULT IS IN THE ON-UNITS
C IN      0: THE FAULT IS TAKEN CARE OF BY FIFAU.(UNIT NOT ON OR BUS)
C         1: THE FAULT IS IN A COMPUTER OR EEM
C
C ***************************************************************************
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/CO12/NTR,IFAU
      COMMON/CO15/NOON(5),NWORK
      COMMON/CO38/IOCU(5),NONDED,NWIO
      COMMON/CO43/IO1
      COMMON/CO46/PCOM,PBU,PBUCO
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSZ,TC
      COMMON/FLTMS/IFLTCT
      INTEGER BUNOND
      INTEGER EXTEN,PTR
      COMMON/COM7/REASON
      INTEGER REASON
      COMMON/FCOUNT/NF(5),NTRF(5)
      DIMENSION IPLACE(5)
C ***************************************************************************
C
      TIME=TABLE(PTR,1)
      DUR=TABLE(PTR,2)
      XMOD=TABLE(PTR,3)
      NUNIT=TABLE(PTR,4)
      MOD=XMOD
      GOTO(1000,2000,1000,3000,1000),MOD
 1000 CONTINUE
      IF(NOON(NUNIT).EQ.1) GOTO 1500
         IN=0
         PTR=PTR+1
         RETURN
 1500 CONTINUE
         CALL EXTENT(EXTEN,DUR,XMOD)
         CALL DETTIM(DETEC,TIME,XMOD)
      CALL COPY(TIME,DETEC,REASON,NEXT)
         NF(NWORK)=NF(NWORK)+1
         IF(DUR.LT.ENDMIS) NTRF(NWORK)=NTRF(NWORK)+1
         IF(ISYNC.EQ.0) CALL ASYNC(TIME,DETEC,NEXT)
         PTR=PTR+1
               IN=1
         RETURN
 2000 CONTINUE
      IF(NONDED.EQ.2) GOTO 2500
         IF(NOON(NUNIT).EQ.1) GOTO 2100
 2050 CONTINUE
         IN=0
         PTR=PTP+1
         RETURN
 2100    CONTINUE
         U=RANF(0.0)
      IF(U.LT.PCOM) GOTO 2110
            IF(DUR.LT.TC) GOTO 2200
```

```
                  CALL BUSCHK(NEXT,NUNIT)
                  IF(NEXT.EQ.5) GOTO 2050
            2110  CONTINUE
                  IF(U.LT.PBU) GOTO 2050
            2200  CONTINUE
                  TABLE(PTR,3)=1
                  GOTO 1500
            2500 CONTINUE
                 CALL IO(NEXT)
                 IN=0
                 RETURN
            3000 CONTINUE
                 IFLTCT=IFLTCT+1
                 CALL UNPACK(NUNIT,IPLACE)
                 CALL FCOFLT(TIME,IPLACE,OUR,NEXT)
                 IN=0
                 PTR=PTR+1
                 RETURN
                 END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | FIFAU | 1 | 34 | 44 | 51 | 66 | 73 | | | | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 14 | | | | | |
| 173 | BUNOND | * INTEGER | *UNDEF | | REFS | 16 | | | | | |
| 2265 | DELAY | REAL | | COM1 | REFS | 8 | | | | | |
| 200 | DETEC | REAL | | | REFS | 37 | 38 | 41 | | | |
| 174 | DUR | REAL | | | REFS | 36 | 40 | 55 | 70 | DEFINED | 25 |
| 74 | ENDMIS | REAL | | COM3 | REFS | 14 | 40 | | | | |
| 2262 | EXTEN | INTEGER | | COM1 | REFS | 8 | 17 | 36 | | | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 8 | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 8 | | | | | |
| 1 | IFAU | INTEGER | | C012 | REFS | 9 | | | | | |
| 0 | IFLTCT | INTEGER | | FLTMS | REFS | 15 | 68 | DEFINED | 68 | | |
| 0 | IN | INTEGER | | F.P. | DEFINED | 1 | 32 | 43 | 49 | 65 | 71 |
| 0 | IOCU | INTEGER | ARRAY | C038 | REFS | 11 | | | | | |
| 0 | IO1 | INTEGER | | C043 | REFS | 12 | | | | | |
| 202 | IPLACE | INTEGER | ARRAY | | REFS | 21 | 69 | 70 | | | |
| 0 | ISYNC | INTEGER | | F.P. | REFS | 41 | DEFINED | 1 | | | |
| 75 | MEMSZ | INTEGER | | COM3 | REFS | 14 | | | | | |
| 177 | MOD | INTEGER | | | REFS | 29 | DEFINED | 28 | | | |
| 0 | NEXT | INTEGER | | F.P. | REFS | 38 | 41 | 56 | 57 | 64 | 70 |
| | | | | | DEFINED | 1 | | | | | |
| 0 | NF | INTEGER | ARRAY | FCOUNT | REFS | 20 | 39 | DEFINED | 39 | | |
| 5 | NONDED | INTEGER | | C038 | REFS | 11 | 46 | | | | |
| 0 | NOON | INTEGER | ARRAY | C015 | REFS | 10 | 31 | 47 | | | |
| 0 | NTR | INTEGER | | C012 | REFS | 9 | | | | | |
| 5 | NTRF | INTEGER | ARRAY | FCOUNT | REFS | 20 | 40 | DEFINED | 40 | | |
| 176 | NUNIT | INTEGER | | | REFS | 31 | 47 | 56 | 69 | DEFINED | 27 |
| 6 | NWOIO | INTEGER | | C038 | REFS | 11 | | | | | |
| 5 | NWORK | INTEGER | | C015 | REFS | 10 | 2*39 | 2*40 | | | |
| 1 | PBU | REAL | | C046 | REFS | 13 | 59 | | | | |
| 2 | PBUCO | REAL | | C046 | REFS | 13 | | | | | |
| 0 | PCOM | REAL | | C046 | REFS | 13 | 54 | | | | |
| 2261 | PTR | INTEGER | | COM1 | REFS | 8 | 17 | 24 | 25 | 26 | 27 | 33 |
| | | | | | | 42 | 50 | 61 | 72 | DEFINED | 33 | 42 | 50 |
| | | | | | | 72 | | | | | | | |
| 0 | REASON | INTEGER | | COM7 | REFS | 18 | 19 | 38 | | | | |
| 2264 | RECOV | REAL | | COM1 | REFS | 8 | | | | | | |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 8 | 24 | 25 | 26 | 27 | | |
| | | | | | DEFINED | 61 | | | | | | |
| 76 | TC | REAL | | COM3 | REFS | 14 | 55 | | | | | |
| 2266 | TIME | REAL | | COM1 | REFS | 8 | 37 | 38 | 41 | 70 | | |
| | | | | | DEFINED | 24 | | | | | | |
| 201 | U | REAL | | | REFS | 54 | 59 | DEFINED | 53 | | | |
| 175 | XHOD | REAL | | | REFS | 28 | 36 | 37 | DEFINED | 26 | | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| ASYNC | | 3 | 41 |
| BUSCHK | | 2 | 56 |
| COPY | | 4 | 38 |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| DETTIM | | 3 | 37 |
| EXTENT | | 3 | 36 |
| FCBFLT | | 4 | 70 |
| IO | | 1 | 64 |
| RANF | REAL | 1 | 53 |
| UNPACK | | 2 | 69 |

| STATEMENT LABELS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|
| 24 | 1000 | 30 | 3*29 | |
| 33 | 1500 | 35 | 31 | 62 |
| 62 | 2000 | 45 | 29 | |
| 70 | 2050 | 48 | 57 | 59 |
| 74 | 2100 | 52 | 47 | |
| 110 | 2110 | 58 | 54 | |
| 112 | 2200 | 60 | 55 | |
| 115 | 2500 | 63 | 46 | |
| 122 | 3000 | 67 | 29 | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 | IDIM | (1) | 1 | TABLE | (1200) | 1201 | PTR | (1) |
| | | 1202 | EXTEN | (1) | 1203 | IDETEC | (1) | 1204 | RECOV | (1) |
| | | 1205 | DELAY | (1) | 1206 | TIME | (1) | | | |
| COI2 | 2 | 0 | NTR | (1) | 1 | IFAU | (1) | | | |
| COI5 | 6 | 0 | NOON | (5) | 5 | NWORK | (1) | | | |
| CO38 | 7 | 0 | IOCU | (5) | 5 | NONDED | (1) | 6 | NHOIO | (1) |
| CO43 | 1 | 0 | IO1 | (1) | | | | | | |
| CO46 | 3 | 0 | PCOM | (1) | 1 | PBU | (1) | 2 | PBUCO | (1) |
| COM3 | 63 | 0 | ACFAU | (60) | 60 | ENDMIS | (1) | 61 | MEMSZ | (1) |
| | | 62 | TC | (1) | | | | | | |
| FLTHS | 1 | 0 | IFLTCT | (1) | | | | | | |
| COM7 | 1 | 0 | REASON | (1) | | | | | | |
| FCOUNT | 10 | 0 | NF | (5) | 5 | NTRF | (5) | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 207B | 135 |
| COMMON LENGTH | 24258 | 1301 |

```
      SUBROUTINE GATHER
C                                        THIS VERSION: 10 APRIL 1974
C THIS SUBROUTINE GATHERS ALL FAULTS ON TOP OF ACFAU
C WE KEEP ALL FAULTS ON TOP OF ACFAU
C REALLY A DELETE AND SQUEEZE THE LIVE/ACTIVE FAULT RECORDS TO THE TOP  SP08APR4
C OF THE ACFAU TABLE.                                                   SP08APR4
C IDIM IS THE MAXIMUM NUMBER OF LURKING FAULTS                          SP08APR4
C
C********** ********* ********* ********* ********* ********* ********
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
C ***************************************************************************
C
   81 CONTINUE
      DO 110 J=1,IDIM
C        FIND ONE THATS NOT ZERO                                        SP08APR4
      IF (ACFAU(J,3).NE.0. )GO TO 110
C        IF ITS AT THE END OF THE TABLE YOU VE DONE YOUR BEST ANYHAY    SP08APR4
      IF (J.EQ.IDIM) RETURN
      K=J+1
C
      DO 120 L=K,IDIM
      IF (ACFAU(L,3).EQ.0.) GO TO 120
C        SO YOU VE GOT A LIVE ONE                                       SP08APR4
      DO 130 I=1,6
C        TRANSFER IT UP TO THE BLANK RECORD                             SP08APR4
      ACFAU(J,I)=ACFAU(L,I)
  130 CONTINUE
C
C        BLANK OUT THE ONE YOU TRANSFERED UP                            SP08APR4
      ACFAU(L,3)=0.
      GO TO 81
  120 CONTINUE
      RETURN
  110 CONTINUE
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | |
|---|---|---|---|---|---|
| 1 | GATHER | 1 | 19 | 34 | 36 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 11 | 17 | 23 | 27 | DEFINED | 27 | 31 |
| 2265 | DELAY | REAL | | COM1 | REFS | 10 | | | | | | |
| 74 | ENDMIS | REAL | | COM3 | REFS | 11 | | | | | | |
| 2262 | EXTEN | REAL | | COM1 | REFS | 10 | | | | | | |
| 40 | I | INTEGER | | | REFS | 2*27 | DEFINED | 25 | | | | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 10 | | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 10 | 15 | 19 | 22 | | | |
| 35 | J | INTEGER | | | REFS | 17 | 19 | 20 | 27 | DEFINED | 15 | |
| 36 | K | INTEGER | | | REFS | 22 | DEFINED | 20 | | | | |
| 37 | L | INTEGER | | | REFS | 23 | 27 | 31 | DEFINED | 22 | | |
| 75 | MEMSIZ | INTEGER | | COM3 | REFS | 11 | | | | | | |
| 2261 | PTR | REAL | | COM1 | REFS | 10 | | | | | | |
| 2264 | RECOV | REAL | | COM1 | REFS | 10 | | | | | | |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 10 | | | | | | |
| 76 | TC | REAL | | COM3 | REFS | 11 | | | | | | |
| 2266 | TIME | REAL | | COM1 | REFS | 10 | | | | | | |

| STATEMENT LABELS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|
| 2 | 81 | 14 | 32 | |
| 32 | 110 | 35 | 15 | 17 |
| 27 | 120 | 33 | 22 | 23 |
| 0 | 130 | 28 | 25 | |

| LOOPS | LABEL | | INDEX | FROM-TO | LENGTH | PROPERTIES | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 110 | * | J | 15 35 | 32B | | EXITS | NOT INNER |
| 15 | 120 | * | L | 22 33 | 15B | | EXITS | NOT INNER |
| 23 | 130 | | I | 25 28 | 2B | INSTACK | | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 IDIM (1) | 1 TABLE (1200) | 1201 PTR (1) | | | |
| | | 1202 EXTEN (1) | 1203 IDETEC (1) | 1204 RECOV (1) | | | |
| | | 1205 DELAY (1) | 1206 TIME (1) | | | | |
| COM3 | 63 | 0 ACFAU (60) | 60 ENDMIS (1) | 61 MEMSIZ (1) | | | |
| | | 62 TC (1) | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 418 | 33 |
| COMMON LENGTH | 2366B | 1270 |

```
      SUBROUTINE GIORF1(TOTAL,VEC,BUS)
      DIMENSION BUS(8)
      DIMENSION VEC(8)
      READ 1000,BUS
      VEC(1)=BUS(1)
      DO 10 I=2,8
         VEC(I)=BUS(I)+VEC(I-1)
   10 CONTINUE
      TOTAL=VEC(8)
      IF(TOTAL.EQ.0) RETURN
      DO 20 I=1,8
        VEC(I)=VEC(I)/TOTAL
   20 CONTINUE
      RETURN
 1000 FORMAT(8F10.5)
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|
| 2 | GIORF1 | 1 | 10 | 14 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BUS | REAL | ARRAY | F.P. | REFS | 2 | 5 | 7 | DEFINED | 1 | 4 |
| 50 | I | INTEGER | | | REFS | 3*7 | 2*12 | DEFINED | 6 | 11 | |
| 0 | TOTAL | REAL | | F.P. | REFS | 10 | 12 | DEFINED | 1 | 9 | |
| 0 | VEC | REAL | ARRAY | F.P. | REFS | 3 | 7 | 9 | 12 | DEFINED | 1 | 5 |
| | | | | | | 7 | 12 | | | | |

| FILE NAMES | MODE | | |
|---|---|---|---|
| INPUT | FMT | READS | 4 |

| STATEMENT LABELS | | | DEF LINE | REFERENCES |
|---|---|---|---|---|
| 0 | 10 | | 8 | 6 |
| 0 | 20 | | 13 | 11 |
| 45 | 1000 | FMT | 15 | 4 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|
| 25 | 10 | I | 6 8 | 38 | INSTACK |
| 37 | 20 | I | 11 13 | 28 | INSTACK |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 708 | 56 |

```
      SUBROUTINE GIORF2(TOTAL,VEC,XM,ISTS,BTUFR)
      DIMENSION VEC(10),ISTS(10),XM(10,4),BTU(10),BTUL(10)
      DIMENSION BTUFR(10,2)
      READ 1000,(BTU(I),I=1,2)
      READ 1000,(BTUL(I),I=1,2)
      READ 1000,(BTU(I),I=3,6)
      READ 1000,(BTUL(I),I=3,6)
      READ 1000,(BTU(I),I=7,10)
      READ 1000,(BTUL(I),I=7,10)
      DO 50 I=1,10
        BTUFR(I,1)=BTU(I)
        BTUFR(I,2)=BTUL(I)
        NO=ISTS(I)+1
        XM(I,1)=BTU(I)
        DO 30 J=2,NO
          XM(I,J)=XM(I,J-1)+BTUL(I)
   30   CONTINUE
        VEC(I)=XM(I,NO)
      IF(VEC(I).EQ.0)GOTO 50
        DO 40 J=1,NO
          XM(I,J)=XM(I,J)/VEC(I)
   40   CONTINUE
   50 CONTINUE
      DO 60 I=2,10
        VEC(I)=VEC(I)+VEC(I-1)
   60 CONTINUE
      TOTAL=VEC(10)
      IF(TOTAL.EQ.0) RETURN
      DO 70 I=1,10
        VEC(I)=VEC(I)/TOTAL
   70 CONTINUE
      RETURN
 1000 FORMAT(8F10.5)
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|
| 2 | GIORF2 | 1 | 28 | 32 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 135 | BTU | REAL | ARRAY | | REFS | 2 | 11 | 14 | DEFINED | 4 | 6 | 8 |
| 0 | BTUFR | REAL | ARRAY | F.P. | REFS | 3 | DEFINED | 1 | 11 | 12 | | |
| 147 | BTUL | REAL | ARRAY | | REFS | 2 | 12 | 16 | DEFINED | 5 | 7 | 9 |
| 132 | I | INTEGER | | | REFS | 4 | 5 | 6 | 7 | 8 | 9 | 2*11 |
| | | | | | 2*12 | 13 | 2*14 | 3*16 | 2*18 | 19 | 3*21 | 3*25 |
| | | | | | 2*30 | DEFINED | 4 | 5 | 6 | 7 | 8 | 9 |
| | | | | | 10 | 24 | 29 | | | | | |
| 0 | ISTS | INTEGER | ARRAY | F.P. | REFS | 2 | 13 | DEFINED | 1 | | | |
| 134 | J | INTEGER | | | REFS | 2*16 | 2*21 | DEFINED | 15 | 20 | | |
| 133 | NO | INTEGER | | | REFS | 15 | 18 | 20 | DEFINED | 13 | | |
| 0 | TOTAL | REAL | | F.P. | REFS | 28 | 30 | DEFINED | 1 | 27 | | |
| 0 | VEC | REAL | ARRAY | F.P. | REFS | 2 | 19 | 21 | 2*25 | 27 | 30 | |
| | | | | | DEFINED | 1 | 18 | 25 | 30 | | | |
| 0 | XM | REAL | ARRAY | F.P. | REFS | 2 | 16 | 18 | 21 | DEFINED | 1 | 14 |
| | | | | | 16 | 21 | | | | | | |

| FILE NAMES | MODE | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| INPUT | FMT | READS | 4 | 5 | 6 | 7 | 8 | 9 |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 30 | | 17 | 15 | | | | | |
| 0 | 40 | | 22 | 20 | | | | | |
| 103 | 50 | | 23 | 10 | 19 | | | | |
| 0 | 60 | | 26 | 24 | | | | | |
| 0 | 70 | | 31 | 29 | | | | | |
| 127 | 1000 | FMT | 33 | 4 | 5 | 6 | 7 | 8 | 9 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|
| 53 | 50 | * I | 10 23 | 338 | NOT INNER |
| 64 | 30 | J | 15 17 | 28 | INSTACK |
| 101 | 40 | J | 20 22 | 28 | INSTACK |
| 107 | 60 | I | 24 26 | 28 | INSTACK |
| 121 | 70 | I | 29 31 | 28 | INSTACK |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 2108 | 136 |

```
      SUBROUTINE GIORF3(TOTAL,VEC,XM,IDEV,IDIM,JDIM,DEV)
      DIMENSION VEC(IDIM),XM(IDIM,JDIM),IDEV(IDIM,JDIM),DEV(JDIM)
      READ 1000,DEV
      DO 30 I=1,IDIM
         XP=0
         DO 10 J=1,JDIM
            XN=XP
            IF(IDEV(I,J).NE.0) XN=XN+DEV(J)
            XP=XN
            XM(I,J)=XN
   10    CONTINUE
         VEC(I)=XN
      IF(XN.EQ.0)GOTO 30
         DO 20 J=1,JDIM
            XM(I,J)=XM(I,J)/XN
   20    CONTINUE
   30 CONTINUE
      DO 40 I=2,IDIM
         VEC(I)=VEC(I)+VEC(I-1)
   40 CONTINUE
      TOTAL=VEC(IDIM)
      IF(TOTAL.EQ.0)RETURN
      DO 50 I=1,IDIM
         VEC(I)=VEC(I)/TOTAL
   50 CONTINUE
      RETURN
 1000 FORMAT(8F10.5)
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | GIORF3 | 1 | 22 26 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | DEV | REAL | ARRAY | F.P. | REFS | 2 | 8 | DEFINED | 1 | 3 | | | |
| 111 | I | INTEGER | | | REFS | 8 | 10 | 12 | 2*15 | 3*19 | 2*24 | | |
| | | | | | DEFINED | 4 | 18 | 23 | | | | | |
| 0 | IDEV | INTEGER | ARRAY | F.P. | REFS | 2 | 8 | DEFINED | 1 | | | | |
| 0 | IDIM | INTEGER | | F.P. | REFS | 3*2 | 4 | 18 | 21 | 23 | | | |
| | | | | | DEFINED | 1 | | | | | | | |
| 113 | J | INTEGER | | | REFS | 2*8 | 10 | 2*15 | DEFINED | 6 | 14 | | |
| 0 | JDIM | INTEGER | | F.P. | REFS | 3*2 | 6 | 14 | DEFINED | 1 | | | |
| 0 | TOTAL | REAL | | F.P. | REFS | 22 | 24 | DEFINED | 1 | 21 | | | |
| 0 | VEC | REAL | ARRAY | F.P. | REFS | 2 | 2*19 | 21 | 24 | DEFINED | 1 | 12 | |
| | | | | | | 19 | 24 | | | | | | |
| 0 | XM | REAL | ARRAY | F.P. | REFS | 2 | 15 | DEFINED | 1 | 10 | 15 | | |
| 114 | XN | REAL | | | REFS | 8 | 9 | 10 | 12 | 13 | 15 | | |
| | | | | | DEFINED | 7 | 8 | | | | | | |
| 112 | XP | REAL | | | REFS | 7 | DEFINED | 5 | 9 | | | | |

| FILE NAMES | MODE | | |
|---|---|---|---|
| INPUT | FMT | READS | 3 |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | |
|---|---|---|---|---|---|
| 0 | 10 | | 11 | 6 | |
| 0 | 20 | | 16 | 14 | |
| 56 | 30 | | 17 | 4 | 13 |
| 0 | 40 | | 20 | 18 | |
| 0 | 50 | | 25 | 23 | |
| 106 | 1000 | FMT | 27 | 3 | |

| LOOPS | LABEL | | INDEX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|---|
| 25 | 30 | * | I | 4 17 | 348 | NOT INNER |
| 33 | 10 | | J | 6 11 | 78 | INSTACK |
| 54 | 20 | | J | 14 16 | 28 | INSTACK |
| 63 | 40 | | I | 18 20 | 28 | INSTACK |
| 77 | 50 | | I | 23 25 | 28 | INSTACK |

STATISTICS

| PROGRAM LENGTH | 1458 | 101 |
|---|---|---|

```
      SUBROUTINE GIORF4(TOTAL,VEC,XM1,XM2,ISTS,DEVS)
      DIMENSION DEVS(4,2)
      DIMENSION VEC(4),XM1(4,4),XM2(4,2),ISTS(2,4),DEV(4),DEVL(4)
      READ 1000,DEV
      READ 1000,DEVL
      DO 20 I=1,4
         DEVS(I,1)=DEV(I)
         DEVS(I,2)=DEVL(I)
         XM1(I,1)=DEV(I)
         DO 10 J=2,4
            XM1(I,J)=XM1(I,J-1)+DEVL(I)
   10    CONTINUE
         IF(ISTS(1,I).EQ.0) XM2(I,1)=0
         IF(ISTS(1,I).NE.0) XM2(I,1)=XM1(I,4)
         IF(ISTS(2,I).EQ.0) XM2(I,2)=XM2(I,1)
         IF(ISTS(2,I).NE.0) XM2(I,2)=XM1(I,4)+XM2(I,1)
         VEC(I)=XM2(I,2)
      IF(XM1(I,4).EQ.0) GOTO 15
      DO 11 J=1,4
      XM1(I,J)=XM1(I,J)/XM1(I,4)
   11 CONTINUE
   15 CONTINUE
      IF(XM2(I,2).EQ.0)GOTO 20
      XM2(I,1)=XM2(I,1)/XM2(I,2)
      XM2(I,2)=1.0
   20 CONTINUE
      DO 30 I=2,4
         VEC(I)=VEC(I)+VEC(I-1)
   30 CONTINUE
      TOTAL=VEC(4)
      IF(TOTAL.EQ.0)RETURN
      DO 40 I=1,4
         VEC(I)=VEC(I)/TOTAL
   40 CONTINUE
      RETURN
 1000 FORMAT(8F10.5)
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | GIORF4 | 1 | 31 35 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 140 | DEV | REAL | ARRAY | | REFS | 3 | 7 | 9 | DEFINED | 4 | | |
| 144 | DEVL | REAL | ARRAY | | REFS | 3 | 8 | 11 | DEFINED | 5 | | |
| 0 | DEVS | REAL | ARRAY | F.P. | REFS | 2 | DEFINED | 1 | 7 | 8 | | |
| 136 | I | INTEGER | | | REFS | 2*7 | 2*8 | 2*9 | 3*11 | 2*13 | 3*14 | 3*15 |
| | | | | | | 4*16 | 2*17 | 18 | 3*20 | 23 | 3*24 | 25 | 3*28 |
| | | | | | | 2*33 | DEFINED | 6 | 27 | 32 | | |
| 0 | ISTS | INTEGER | ARRAY | F.P. | REFS | 3 | 13 | 14 | 15 | 16 | | |
| | | | | | DEFINED | 1 | | | | | | |
| 137 | J | INTEGER | | | REFS | 2*11 | 2*20 | DEFINED | 10 | 19 | | |
| 0 | TOTAL | REAL | | F.P. | REFS | 31 | 33 | DEFINED | 1 | 30 | | |
| 0 | VEC | RFAL | ARRAY | F.P. | REFS | 3 | 2*28 | 30 | 33 | DEFINED | 1 | 17 |
| | | | | | | 28 | 33 | | | | | |
| 0 | XM1 | REAL | ARRAY | F.P. | REFS | 3 | 11 | 14 | 16 | 18 | 2*20 | |
| | | | | | DEFINED | 1 | 9 | 11 | 20 | | | |
| 0 | XM2 | REAL | ARRAY | F.P. | REFS | 3 | 15 | 16 | 17 | 23 | 2*24 | |
| | | | | | DEFINED | 1 | 13 | 14 | 15 | 16 | 24 | 25 |

| FILE NAMES | MODE | | |
|---|---|---|---|
| INPUT | FMT | READS | 5 |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | |
|---|---|---|---|---|---|
| 0 | 10 | | 12 | 10 | |
| 0 | 11 | | 21 | 19 | |
| 77 | 15 | | 22 | 18 | |
| 106 | 20 | | 26 | 6 | 23 |
| 0 | 30 | | 29 | 27 | |
| 0 | 40 | | 34 | 32 | |
| 132 | 1000 | FMT | 36 | 4 | 5 |

| LOOPS | LABEL | | INDEX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|---|
| 27 | 20 | * | I | 6 26 | 628 | NOT INNER |
| 35 | 10 | | J | 10 12 | 28 | INSTACK |
| 75 | 11 | | J | 19 21 | 28 | INSTACK |
| 112 | 30 | | I | 27 29 | 28 | INSTACK |
| 124 | 40 | | I | 32 34 | 28 | INSTACK |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 220B | 144 |

```
      LOGICAL FUNCTION GREATR(VAL)
      GREATR=RANF(0.).GT.VAL
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | GREATR | 1 | 3 |

| VARIABLES | SN | TYPE | RELOCATION | | | | |
|---|---|---|---|---|---|---|---|
| 14 | GREATR | LOGICAL | | DEFINED | 2 | | |
| 0 | VAL | REAL | F.P. | PEFS | 2 | DEFINED | 1 |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| RANF | REAL | 1 | 2 |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 158 | 13 |

```
      SUBROUTINE INCOPY(NC,FIELDA,FIELDB)
      INTEGER FIELDA,FIELDB
      DIMENSION FIELDA(NC),FIELDB(NC)
      DO 10 I=1,NC
5       FIELDA(I)=FIELDB(I)
     10 CONTINUE
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | INCOPY | 1 | 7 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | FIELDA | INTEGER | ARRAY | F.P. | REFS | 2 | 3 | DEFINED | 1 | 5 | |
| 0 | FIELDB | INTEGER | ARRAY | F.P. | REFS | 2 | 3 | 5 | DEFINED | 1 | |
| 24 | I | INTEGER | | | REFS | 2*5 | DEFINED | 4 | | | |
| 0 | NC | INTEGER | | F.P. | REFS | 2*3 | 4 | DEFINED | 1 | | |

| STATEMENT LABELS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 0 | 10 | 6 | 4 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|
| 17 | 10 | I | 4 6 | 28 | INSTACK |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 348 | 28 |

```
      SUBROUTINE INFLTB
      COMMON/TAB/TABLE(100)
      DIMENSION IPLACE(5)
      INTEGER TRANS,PERM,TABLE
      DATA TRANS/1HT/,PERM/1HP/
      DATA NH/1HN/
      J=1
   20 CONTINUE
      KJ=0
      READ 2000,ITYP,(IPLACE(M),M=2,5)
      IF(ITYP.EQ.TRANS) KJ=1
      IF(ITYP.EQ.PERM) KJ=2
      IPLACE(1)=KJ
      PRINT 2001,IPLACE
      CALL PACK(IPLACE,LOC)
      TABLE(J)=LOC
      J=J+1
      IF((ITYP.EQ.NH).OR.(KJ.NE.0)) GOTO 20
      TABLE(J)=-1
 2000 FORMAT(A1,4X,4I5)
 2001 FORMAT(1X,5I5)
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 1 | INFLTB | 1 | 22 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 64 | IPLACE | INTEGER | ARRAY | | REFS | 3 | 14 | 15 | DEFINED | 10 | 13 | |
| 61 | ITYP | INTEGER | | | REFS | 11 | 12 | 18 | DEFINED | 10 | | |
| 57 | J | INTEGER | | | REFS | 16 | 17 | 19 | DEFINED | 7 | 17 | |
| 60 | KJ | INTEGER | | | REFS | 13 | 18 | DEFINED | 9 | 11 | 12 | |
| 63 | LOC | INTEGER | | | REFS | 15 | 16 | | | | | |
| 62 | M | INTEGER | | | REFS | 10 | DEFINED | 10 | | | | |
| 52 | NW | INTEGER | | | REFS | 18 | DEFINED | 6 | | | | |
| 51 | PERM | INTEGER | | | REFS | 4 | 12 | DEFINED | 5 | | | |
| 0 | TABLE | INTEGER | ARRAY | TAB | REFS | 2 | 4 | DEFINED | 16 | 19 | | |
| 50 | TRANS | INTEGER | | | REFS | 4 | 11 | DEFINED | 5 | | | |

| FILE NAMES | MODE | | |
|---|---|---|---|
| INPUT | FMT | READS | 10 |
| OUTPUT | FMT | WRITES | 14 |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| PACK | | 2 | 15 |

| STATEMENT LABELS | | | DEF LINE | REFERENCES |
|---|---|---|---|---|
| 3 | 20 | | 8 | 18 |
| 53 | 2000 | FMT | 20 | 10 |
| 55 | 2001 | FMT | 21 | 14 |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) |
|---|---|---|
| TAB | 100 | 0 TABLE (100) |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 718 | 57 |
| COMMON LENGTH | 144B | 100 |

```
      SUBROUTINE IO(NEXT)
C                                        THIS VERSION: 12 JULY 1974
C THIS SUBROUTINE  RETURNS NEXT=5 IF ALL EEMS  ARE LOST.
C
C ******************************************************************
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM3/A(10,6),ENDMIS,M,TC
      COMMON/COM7/REASON
      COMMON/CO12/NTR,IFAU
      COMMON/CO16/NDIAG,NUNDI
      COMMON/CO36/RMISTH
      COMMON/CO38/IOCU(5),NONDED,NWOIO
      COMMON/CO39/PDETIO
      COMMON/CO40/ISWI,FSWI,NSWION,NIO
      INTEGER PTR,EXTEN,REASON
C ******************************************************************
C
      I=TABLE(PTR,4)
      PTR=PTR+1
C COUNTING
      IFAU=IFAU+1
      GO TO (4,2,3,3,3),NWOIO
    4 CONTINUE
      REASON=2
      NEXT=5
      RETURN
C LOOK IF WE CAN ISOLATE FAULT.
    2 CONTINUE
      IF (RANF(0.).GT.PDETIO) GO TO 4
    3 CONTINUE                                                  L1580IO
      IF (TABLE(PTR-1,2).LT.TC) GO TO 1
      NWOIO=NWOIO-1
      IOCU(I)=0
    1 CONTINUE
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | IO | 1 | 26 35 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | A | REAL | ARRAY | COM3 | REFS | 7 | | | | | |
| 2265 | DELAY | REAL | | COM1 | REFS | 6 | | | | | |
| 74 | ENDMIS | REAL | | COM3 | REFS | 7 | | | | | |
| 2262 | EXTEN | INTEGER | | COM1 | REFS | 6 | 15 | | | | |
| 1 | FSWI | REAL | | C040 | REFS | 14 | | | | | |
| 43 | I | INTEGER | | | REFS | 33 | DEFINED | 18 | | | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 6 | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 6 | | | | | |
| 1 | IFAU | INTEGER | | C012 | REFS | 9 | 21 | DEFINED | 21 | | |
| 0 | IOCU | INTEGER | ARRAY | C038 | REFS | 12 | DEFINED | 33 | | | |
| 0 | ISWI | INTEGER | | C040 | REFS | 14 | | | | | |
| 75 | M | INTEGER | | COM3 | REFS | 7 | | | | | |
| 0 | NDIAG | INTEGER | | C016 | REFS | 10 | | | | | |
| 0 | NEXT | INTEGER | | F.P. | DEFINED | 1 | 25 | | | | |
| 3 | NIO | INTEGER | | C040 | REFS | 14 | | | | | |
| 5 | NONDED | INTEGER | | C036 | REFS | 12 | | | | | |
| 2 | NSWION | INTEGER | | C040 | REFS | 14 | | | | | |
| 0 | NTR | INTEGER | | C012 | REFS | 9 | | | | | |
| 1 | NUNDI | INTEGER | | C016 | REFS | 10 | | | | | |
| 6 | NWDIO | INTEGER | | C038 | REFS | 12 | 22 | 32 | DEFINED | 32 | |
| 0 | POETIO | REAL | | C039 | REFS | 13 | 29 | | | | |
| 2261 | PTR | INTEGER | | COM1 | REFS | 6 | 15 | 18 | 19 | 31 | |
| | | | | | DEFINED | 19 | | | | | |
| 0 | REASON | INTEGER | | COM7 | REFS | 8 | 15 | DEFINED | 24 | | |
| 2264 | RECOV | REAL | | COM1 | REFS | 6 | | | | | |
| 0 | RMISTM | REAL | | C036 | REFS | 11 | | | | | |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 6 | 18 | 31 | | | |
| 76 | TC | REAL | | COM3 | REFS | 7 | 31 | | | | |
| 2266 | TIME | REAL | | COM1 | REFS | 6 | | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| RANF | REAL | 1 | 29 |

| STATEMENT LABELS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 36 | 1 | 34 | 31 |
| 25 | 2 | 28 | 22 |
| 31 | 3 | 30 | 3*22 |
| 21 | 4 | 23 | 22 29 |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS | NAME | (LENGTH) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 | IDIM | (1) | 1 | TABLE | (1200) | 1201 | PTR | (1) | |
| | | 1202 | EXTEN | (1) | 1203 | IDETEC | (1) | 1204 | RECOV | (1) | |
| | | 1205 | DELAY | (1) | 1206 | TIME | (1) | | | | |
| COM3 | 63 | 0 | A | (60) | 60 | ENDMIS | (1) | 61 | M | (1) | |
| | | 62 | TC | (1) | | | | | | | |
| COM7 | 1 | 0 | REASON | (1) | | | | | | | |
| C012 | 2 | 0 | NTR | (1) | 1 | IFAU | (1) | | | | |
| C016 | 2 | 0 | NDIAG | (1) | 1 | NUNDI | (1) | | | | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME (LENGTH) | | |
|---|---|---|---|---|
| C036 | 1 | 0 RMISTM (1) | | |
| C038 | 7 | 0 IOCU (5) | 5 NONDED (1) | 6 NWOIO (1) |
| C039 | 1 | 0 PDETIO (1) | | |
| C040 | 4 | 0 ISHI (1) | 1 FSHI (1) | 2 NSWION (1) |
| | | 3 NIO (1) | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 448 | 36 |
| COMMON LENGTH | 24108 | 1288 |

```
      FUNCTION IRAN(MIN,MAX)
C                                     THIS VERSION: 25 FEBRUARY 1974
C IT RETURNS A RANDOM INTEGER BETWEEN MIN AND MAX(INCLUDED)
C********** ********* ********* ********** ********* ********* ********
5     C
      IRAN=INT((MAX-MIN+1)*RANF(0.))+MIN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | IRAN | 1 | 7 |

| VARIABLES | SN | TYPE | RELOCATION | | | | |
|---|---|---|---|---|---|---|---|
| 17 | IRAN | INTEGER | | DEFINED | 6 | | |
| 0 | MAX | INTEGER | F.P. | REFS | 6 | DEFINED | 1 |
| 0 | MIN | INTEGER | F.P. | REFS | 2*6 | DEFINED | 1 |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| RANF | REAL | 1 | 6 |

| INLINE FUNCTIONS | TYPE | ARGS | | DEF LINE | REFERENCES |
|---|---|---|---|---|---|
| INT | INTEGER | 1 | INTRIN | | 6 |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 208 | 16 |

```
      FUNCTION ISTEPD(N,VEC)
      DIMENSION VEC(N)
      U=RANF(0.0)
      DO 10 J=1,N
        IF(U.LE.VEC(J)) GOTO 20
   10 CONTINUE
      STOP 4005
   20 CONTINUE
      ISTEPD=J
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | ISTEPD | 1 | 10 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 40 | ISTEPD | INTEGER | | | DEFINED | 9 | | | | |
| 42 | J | INTEGER | | | REFS | 5 | 9 | DEFINED | 4 | |
| 0 | N | INTEGER | | F.P. | REFS | 2 | 4 | DEFINED | 1 | |
| 41 | U | REAL | | | REFS | 5 | DEFINED | 3 | | |
| 0 | VEC | REAL | ARRAY | F.P. | REFS | 2 | 5 | DEFINED | 1 | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| RANF | REAL | 1 | 3 |

| STATEMENT LABELS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 0 | 10 | 6 | 4 |
| 31 | 20 | 8 | 5 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | EXITS |
|---|---|---|---|---|---|---|
| 23 | 10 | * J | 4 6 | 40 | INSTACK | |

STATISTICS

PROGRAM LENGTH 528 42

```
      LOGICAL FUNCTION LESS(VAL)
C
C THIS PROGRAM GENERATES A UNIFORM PANDOM NUMBER BETWEEN ZERO AND ONE,
C AND THEN DETERMINES IF IT IS LESS THAN A GIVEN VALUE.  IF IT IS,
C .TRUE. IS RETURNED; ELSE .FALSE. IS RETURNED.  THIS FUNCTION IS USED
C TO DETERMINE THE SUCCESS OF FAULT RECOVERY
C
C VAL    THE GIVEN VALUE.----SUCH AS A TRANSIENTE LEAKAGE
C
C*************************************************************************
C
      U=RANF(0.)
      LESS = U.LE.VAL
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | LESS | 1 | 14 |

| VARIABLES | SN | TYPE | RELOCATION | | | | |
|---|---|---|---|---|---|---|---|
| 15 | LESS | LOGICAL | | DEFINED | 13 | | |
| 16 | U | REAL | | REFS | 13 | DEFINED | 12 |
| 0 | VAL | REAL | F.P. | REFS | 13 | DEFINED | 1 |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| RANF | REAL | 1 | 12 |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 178 | 15 |

```
      SUBROUTINE  MDMFLT(MDM,PRT)
C
C THIS ROUTINE IS INVOKED WHEN A FAULT OCCURS IN A BUS TERMINAL UNIT
C (AN MDM OR DDU).  IT DETERMINES THE EFFECT OF THIS FAULT ON THE
C FLIGHT CRITICAL BUS EQUIPMENT GROUP.
C
C MDM    IDENTIFIES THE FAULTY MDM
C PRT    IF NONZERO, INDICATES ONLY A PORT FAILED AND IDENTIFIES IT
C
C*********************************************************************
      COMMON/FCB1/BUSSTS(8),FCB(8,10),BTUSTS(10)
      COMMON/FCB2/NBTU,BTUTYP(10),BTUCON(10,4),BTUNO(10)
      COMMON/FCB5/BTUCOV(10,2),BTUTFL(10,2)
      COMMON/FCB6/NBTUTF(10,2),NBTUPF(10,2),NBTUTR(10,2)
      COMMON/CO12/NTR,IFAU
      COMMON/STATUS/STS(20)
      COMMON/FAULT/FLTTYP
      COMMON/FCBUC/FCBUCF(6)
      INTEGER BTYP,BUS,BUSSTS,FCB,BTUSTS
      COMMON/FLAGS/FLG(5)
      LOGICAL FLG
      INTEGER FCBUCF
      LOGICAL GREATR
      INTEGER BTUTYP,BTUCON,BTUNO,STS,FLTTYP
      INTEGER PRT
C
      IF(BTUSTS(MDM).EQ.0) RETURN
      FLG(1)=.TRUE.
      BTYP = BTUTYP(MDM)
      IF(PRT.EQ.0) GOTO 500
C
C                 ...THE FAULT OCCURED IN ONE OF THE BTU PORTS
      BUS=BTUCON(MDM,PRT)
      IF(BUS.EQ.0)STOP 1002
      ITYP=FCB(BUS,MDM)
      IF(ITYP.EQ.0) RETURN
C                 ...INCREMENT FAULT COUNTER AND JUMP ON FAULT TYPE
      IFAU = IFAU+1
      GOTO(100,200),FLTTYP
C
C                 ...ITS A TRANSIENT
  100 CONTINUE
      NTR = NTR+1
      NBTUTF(BTYP,1)=NBTUTF(BTYP,1)+1
C                 ...IF TRANSIENT RECOVERY FAILS, GOTO PERMANENT RECOV
      IF(ITYP.EQ.2)RETURN
      IF(GREATR(BTUTFL(MDM,1))) RETURN
      NBTUTR(BTYP,1)=NBTUTR(BTYP,1)+1
      GOTO 205
C
C                 ...ITS A PERMANENT
  200 CONTINUE
      NBTUPF(BTYP,1) = NBTUPF(BTYP,1) + 1
C                 ...ENTER---LEAKY TRANSIENT
  205 CONTINUE
```

```
      GOTO (300,400),ITYP
C
C                 ...AN ACTIVE PORT HAS FAILED
  300 CONTINUE
      FCB(BUS,MDM)=0
      NL=BTUSTS(MDM)-1
      BTUSTS(MDM)=NL
      IF(NL.EQ.0) GOTO 350
      IF(BTUTYP(MDM).EQ.1) RETURN
C                 ...IF FAULT COVERED, FIND BACKUP PORT
      IF(GREATR(BTUCOV(MDM,1))) GOTO 900
      DO  330 I=1,2
        IBUS=BTUCON(MDM,I)
        IF(FCB(IBUS,MDM).EQ.2) GOTO 335
  330 CONTINUE
C                 ...SOMETHINGS WRONG WITH FCB.
      PRINT 1000,BTUCON
 1000 FORMAT(/12H ***ERROR***/1X,4(10I5/)/)
      CALL PIOCNF
C                 ...FOUND THE BACKUP PORT
  335 FCB(IBUS,MDM)=1
      RETURN
C
C                 ...NO BACKUP PORTS ARE LEFT
  350 CONTINUE
      CALL MDMPF(MDM)
      IF(STS(1).NE.0) CALL SETSTS(24+BTUTYP(MDM))
      RETURN
C
C                 ...THE BACKUP PORT FAILED
  400 CONTINUE
      FCB(BUS,MDM)=0
      BTUSTS(MDM)=BTUSTS(MDM)-1
      RETURN
C
C                 ...THE WHOLE BTU FAILED
  500 CONTINUE
      IFAU = IFAU+1
      GOTO(600,700),FLTTYP
C
C                 ...ITS ONLY A TRANSIENT
  600 CONTINUE
      NTR=NTR+1
      NBTUTF(BTYP,2) = NBTUTF(BTYP,2)+1
C                 ...IS TRANSIENT RECOVERY SUCCESSFUL
      IF(GREATR(BTUTFL(MDM,2))) RETURN
      NBTUTR(BTYP,2) = NBTUTR(BTYP,2)+1
      GOTO 705
C
C                 ...THE FAULT IS PERMANENT
  700 CONTINUE
      NBTUPF(BTYP,2)=NBTUPF(BTYP,2)+1
C                 ...ENTER LEAKY TRANSIENTS
  705 CONTINUE
      IF(GREATR(BTUCOV(MDM,2))) GOTO 900
```

```
      C                 ...REMOVE MDM FROM SYSTEM
            DO 710 I=1,3
              IBUS=BTUCON(MDM,I)
            IF(IBUS.EQ.0) GOTO 750
115           FCB(IBUS,MDM)=0
        710 CONTINUE
      C
      C                 ...DETERMINE EFFECT OF FAULT
        750 CONTINUE
120     BTUSTS(MDM)=0
        CALL  MDMPF(MDM)
        IF(STS(1).NE.0) CALL SETSTS(24+BTUTYP(MDM))
        RETURN
      C
125   C                 ...AN UNCOVERED FAULT OCCURED
        900 CONTINUE
        CALL SETSTS(FCBUCF(2))
        RETURN
        END
```

SYMBOLIC REFERENCE MAP

ENTRY POINTS DEF LINE REFERENCES

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | MDMFLT | 1 | 27 | 36 | 46 | 47 | 64 | 77 | 83 | 89 | 101 | 123 |
| | | | 128 | | | | | | | | |

VARIABLES SN TYPE RELOCATION

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | BTUCON | INTEGER | ARRAY | FCB2 | REFS | 12 | 24 | 33 | 68 | 72 | 113 | |
| 0 | BTUCOV | REAL | ARRAY | FCB5 | REFS | 13 | 66 | 110 | | | | |
| 63 | BTUNO | INTEGER | ARRAY | FCB2 | REFS | 12 | 24 | | | | | |
| 130 | BTUSTS | INTEGER | ARRAY | FCB1 | REFS | 11 | 19 | 27 | 61 | 88 | | |
| | | | | | DEFINED | 62 | 88 | 120 | | | | |
| 24 | BTUTFL | REAL | ARRAY | FCB5 | REFS | 13 | 47 | 101 | | | | |
| 1 | BTUTYP | INTEGER | ARRAY | FCB2 | REFS | 12 | 24 | 29 | 64 | 82 | 122 | |
| 276 | BTYP | INTEGER | | | REFS | 19 | 2*44 | 2*48 | 2*53 | 2*99 | 2*102 | 2*107 |
| | | | | | DEFINED | 29 | | | | | | |
| 277 | BUS | INTEGER | | | REFS | 19 | 34 | 35 | 60 | 87 | | |
| | | | | | DEFINED | 33 | | | | | | |
| 0 | BUSSTS | INTEGER | ARRAY | FCB1 | REFS | 11 | 19 | | | | | |
| 10 | FCB | INTEGER | ARRAY | FCB1 | REFS | 11 | 19 | 35 | 69 | DEFINED | 60 | 76 |
| | | | | | 87 | 115 | | | | | | |
| 0 | FCBUCF | INTEGER | ARRAY | FCRUC | REFS | 18 | 22 | 127 | | | | |
| 0 | FLG | LOGICAL | ARRAY | FLAGS | REFS | 20 | 21 | DEFINED | 28 | | | |
| 0 | FLTTYP | INTEGER | | FAULT | REFS | 17 | 24 | 39 | 94 | | | |
| 302 | I | INTEGER | | | REFS | 68 | 113 | DEFINED | 67 | 112 | | |
| 303 | IBUS | INTEGER | | | REFS | 69 | 76 | 114 | 115 | DEFINED | 68 | 113 |
| 1 | IFAU | INTEGER | | C012 | REFS | 15 | 38 | 93 | DEFINED | 38 | 93 | |
| 300 | ITYP | INTEGER | | | REFS | 36 | 46 | 56 | DEFINED | 35 | | |
| 0 | MDM | INTEGER | | F.P. | REFS | 27 | 29 | 33 | 35 | 47 | 60 | 61 |
| | | | | | 62 | 64 | 66 | 68 | 69 | 76 | 81 | 82 |
| | | | | | 87 | 2*88 | 101 | 110 | 113 | 115 | 120 | 121 |
| | | | | | 122 | DEFINED | 1 | | | | | |
| 0 | NBTU | INTEGER | | FCB2 | REFS | 12 | | | | | | |
| 24 | NBTUPF | INTEGER | ARRAY | FCB6 | REFS | 14 | 53 | 107 | DEFINED | 53 | 107 | |
| 0 | NBTUTF | INTEGER | ARRAY | FCB6 | REFS | 14 | 44 | 99 | DEFINED | 44 | 99 | |
| 50 | NBTUTR | INTEGER | ARRAY | FCB6 | REFS | 14 | 48 | 102 | DEFINED | 48 | 102 | |
| 301 | NL | INTEGER | | | REFS | 62 | 63 | DEFINED | 61 | | | |
| 0 | NTR | INTEGER | | C012 | REFS | 15 | 43 | 98 | DEFINED | 43 | 98 | |
| 0 | PRT | INTEGER | | F.P. | REFS | 25 | 30 | 33 | DEFINED | 1 | | |
| 0 | STS | INTEGER | ARRAY | STATUS | REFS | 16 | 24 | 82 | 122 | | | |

FILE NAMES MODE

| | | | | |
|---|---|---|---|---|
| OUTPUT | FMT | WRITES | 72 | |

EXTERNALS TYPE ARGS REFERENCES

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| GREATR | LOGICAL | 1 | 23 | 47 | 66 | 101 | 110 |
| MDMPF | | 1 | 81 | 121 | | | |
| PIOCNF | | 0 | 74 | | | | |
| SETSTS | | 1 | 82 | 122 | 127 | | |

STATEMENT LABELS DEF LINE REFERENCES

| | | | |
|---|---|---|---|
| 55 | 100 | 42 | 39 |
| 74 | 200 | 52 | 39 |
| 76 | 205 | 55 | 49 |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | |
|---|---|---|---|---|---|
| 104 | 300 | | 59 | 56 | |
| 0 | 330 | | 70 | 67 | |
| 145 | 335 | | 76 | 69 | |
| 151 | 350 | | 80 | 63 | |
| 165 | 400 | | 86 | 56 | |
| 172 | 500 | | 92 | 30 | |
| 202 | 600 | | 97 | 94 | |
| 215 | 700 | | 106 | 94 | |
| 217 | 705 | | 109 | 103 | |
| 0 | 710 | | 116 | 112 | |
| 237 | 750 | | 119 | 114 | |
| 254 | 900 | | 126 | 66 | 110 |
| 270 | 1000 | FMT | 73 | 72 | |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | |
|---|---|---|---|---|---|---|
| 130 | 330 | * I | 67 70 | 68 | INSTACK | EXITS |
| 230 | 710 | * I | 112 116 | 78 | INSTACK | EXITS |

| COMMON BLOCKS | LENGTH | MEMBERS | - BIAS | NAME | (LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| FCB1 | 98 | | 0 | BUSSTS | (8) | | 8 | FCB | (80) | 88 BTUSTS (10) |
| FCB2 | 61 | | 0 | NBTU | (1) | | 1 | BTUTYP | (10) | 11 BTUCON (40) |
| | | | 51 | BTUNO | (10) | | | | | |
| FCB5 | 40 | | 0 | BTUCOV | (20) | | 20 | BTUTFL | (20) | |
| FCB6 | 60 | | 0 | NBTUTF | (20) | | 20 | NBTUPF | (20) | 40 NBTUTR (20) |
| C012 | 2 | | 0 | NTR | (1) | | 1 | IFAU | (1) | |
| STATUS | 20 | | 0 | STS | (20) | | | | | |
| FAULT | 1 | | 0 | FLTTYP | (1) | | | | | |
| FCBUC | 6 | | 0 | FCBUCF | (6) | | | | | |
| FLAGS | 5 | | 0 | FLG | (5) | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 3138 | 203 |
| COMMON LENGTH | 4458 | 293 |

```
      SUBROUTINE MDMPF(IIMDM)
C
C THIS ROUTINE DETERMINS THE EFFECT OF THE LOSS OF AN MDM OR DDU
C FAULT ON THE FLIGT CRITICAL DEVICES.  IT IS INVOKED BY BUSFLT
C OR MDMFLT.
C
C MDM    IDENTIFIES THE FAULTY MDM OR DDU.
C
C*********************************************************************
      COMMON/FCB2/NBTU,BTUTYP(10),BTUCON(10,4),BTUNO(10)
      COMMON/FCB7/NDDUDV,DDUDVS(4),DDUDV(2,4),DDUDVN(4)
      COMMON/FCB8/NDFFDV,DFFDVS(6),DFFDV(4,6),DFFDVN(6)
      COMMON/FCB9/NNFFDV,NFFDVS(2,4),NFFDV(3,2,4),NFFDVN(4)
      COMMON/FCB10/NDFADV,DFADVS(3),DFADV(4,3),DFADVN(3)
      COMMON/STATUS/STS(20)
      COMMON/FLAGS/FLG(5)
      LOGICAL FLG
      INTEGER BTUTYP,BTUCON,BTUNO,DDUDVS,DDUDV,DDUDVN
      INTEGER DFFDVS,DFFDV,DFFDVN,DFADVS,DFADV,DFADVN,STS,DDUNO
      MDM=IIMDM
C
C               ...IS THE BTU A DDU, AN FF MDM, OR A FA MDM.
      MDMTYP = BTUTYP(MDM)
      GOTO(100,200,300),MDMTYP
C
C               ...A DDU HAS FAILED
  100 CONTINUE
      FLG(2)=.TRUE.
      DDUNO = BTUNO(MDM)
      DO 120 I=1,NDDUDV
        IF(DDUDV(DDUNO,I).EQ.0)GOTO 120
        DDUDV(DDUNO,I)=0
        NL=DDUDVS(I)-1
        DDUDVS(I)=NL
        IF(NL.NE.0)GOTO 120
C               ...A FUNCTION IS NO LONGER AVAILABLE
        CALL SETSTS(DDUDVN(I))
  120 CONTINUE
C               ...THE CORRESPONDING PILOT CONTROLS ARE ALSO DISABLED
      FLG(4)=.TRUE.
      DO 130 I=2,NNFFDV
        IF(NFFDVS(DDUNO,I).EQ.0) GOTO 130
        NFFDVS(DDUNO,I)=0
      IF(NFFDVS(3-DDUNO,I).NE.0) GOTO 130
C               ...A PILOT CONTROL FUNCTION HAS FAILED
        CALL SETSTS(NFFDVN(I))
  130 CONTINUE
      RETURN
C
C               ...A FAULT OCCURS IN THE FLIGHT FORWARD MDMS
  200 CONTINUE
      FLG(3)=.TRUE.
      FLG(4)=.TRUE.
      MDM=BTUNO(MDM)
      DO 220 I=1,NDFFDV
```

```
                IF(DFFDV(MDM,I).EQ.0)GOTO 220
                DFFDV(MDM,I)=0
                NL=DFFDVS(I)-1
                DFFDVS(I)=NL
60              IF(NL.NE.0) GOTO 220
      C                    ...A FUNCTION IS NO LONGER AVAILABLE
                CALL SETSTS(DFFDVN(I))
            220 CONTINUE
      C                 ...CHECK THE NON-DEDICATED DEVICES
65              IF(MDM.GE.4) RETURN
                DO 250 I=1,2
                  DO 230 J=1,NNFFDV
                    IF( NFFDVS(I,J).EQ.0) GOTO 230
                    IF(NFFDV(MDM,I,J).EQ.0) GOTO 230
70                  NFFDV(MDM,I,J)=0
                    NL=NFFDVS(I,J)-1
                    NFFDVS(I,J)=NL
                    IF(NL.NE.0) GOTO 230
      C                 ...CHECK TO SEE THAT THE CORRESPONDING DEVICE IS OK
75              IF(NFFDVS(3-I,J).NE.0) GOTO 230
                    CALL SETSTS(NFFDVN(J))
            230   CONTINUE
            250 CONTINUE
                RETURN
80        C
          C                 ...AN FA-MDM HAS FAILED
            300 CONTINUE
                FLG(5)=.TRUE.
                MDM=BTUNO(MDM)
85              DO 320 I=1,NOFADV
                  IF(DFADV(MDM,I).EQ.0) GOTO 320
                  DFADV(MDM,I)=0
                  NL=DFADVS(I)-1
                  DFADVS(I)=NL
90                IF(NL.GT.1) GOTO 320
                  IF((NL.NE.0).AND.(I.NE.1)) GOTO 320
                  DFADVS(I)=0
                  CALL SETSTS(DFADVN(I))
            320 CONTINUE
95              RETURN
                END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | MOMPF | 1 | 48 | 65 | 79 | 95 | | | | | |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | BTUCON | INTEGER | ARRAY | FCB2 | REFS | 10 | 18 | | | | |
| 63 | BTUNO | INTEGER | ARRAY | FCB2 | REFS | 10 | 18 | 29 | 54 | 84 | |
| 1 | BTUTYP | INTEGER | ARRAY | FCB2 | REFS | 10 | 18 | 23 | | | |
| 5 | DDUDV | INTEGER | ARRAY | FCB7 | REFS | 11 | 18 | 31 | DEFINED | 32 | |
| 15 | DDUDVN | INTEGER | ARRAY | FCB7 | REFS | 11 | 18 | 37 | | | |
| 1 | DDUDVS | INTEGER | ARRAY | FCB7 | REFS | 11 | 18 | 33 | DEFINED | 34 | |
| 221 | DDUNO | INTEGER | | | REFS | 19 | 31 | 32 | 42 | 43 | 44 |
| | | | | | DEFINED | 29 | | | | | |
| 4 | DFADV | INTEGER | ARRAY | FCB10 | REFS | 14 | 19 | 86 | DEFINED | 87 | |
| 20 | DFADVN | INTEGER | ARRAY | FCB10 | REFS | 14 | 19 | 93 | | | |
| 1 | DFADVS | INTEGER | ARRAY | FCB10 | REFS | 14 | 19 | 88 | DEFINED | 89 | 92 |
| 7 | DFFDV | INTEGER | ARRAY | FCB8 | REFS | 12 | 19 | 56 | DEFINED | 57 | |
| 37 | DFFDVN | INTEGER | ARRAY | FCB8 | REFS | 12 | 19 | 62 | | | |
| 1 | DFFDVS | INTEGER | ARRAY | FCB8 | REFS | 12 | 19 | 58 | DEFINED | 59 | |
| 0 | FLG | LOGICAL | ARRAY | FLAGS | REFS | 16 | 17 | DEFINED | 28 | 40 | 52 |
| | | | | | 83 | | | | | | 53 |
| 224 | I | INTEGER | | | REFS | 31 | 32 | 33 | 34 | 37 | 42 |
| | | | | | 44 | 46 | 56 | 57 | 58 | 59 | 62 |
| | | | | | 69 | 70 | 71 | 72 | 75 | 86 | 87 |
| | | | | | 89 | 91 | 92 | 93 | DEFINED | 30 | 41 |
| | | | | | 66 | 85 | | | | | 55 |
| 0 | IIMOM | INTEGER | | F.P. | REFS | 20 | DEFINED | 1 | | | |
| 226 | J | INTEGER | | | REFS | 68 | 69 | 70 | 71 | 72 | 75 |
| | | | | | DEFINED | 67 | | | | | 76 |
| 222 | MOM | INTEGER | | | REFS | 23 | 29 | 54 | 56 | 57 | 65 |
| | | | | | 70 | 84 | 86 | 87 | DEFINED | 20 | 54 |
| 223 | MOMTYP | INTEGER | | | REFS | 24 | DEFINED | 23 | | | 84 |
| 0 | NBTU | INTEGER | | FCB2 | REFS | 10 | | | | | |
| 0 | NDDUDV | INTEGER | | FCB7 | REFS | 11 | 30 | | | | |
| 0 | NDFADV | INTEGER | | FCB10 | REFS | 14 | 85 | | | | |
| 0 | NDFFDV | INTEGER | | FCB8 | REFS | 12 | 55 | | | | |
| 11 | NFFDV | INTEGER | ARRAY | FCB9 | REFS | 13 | 69 | DEFINED | 70 | | |
| 41 | NFFDVN | INTEGER | ARRAY | FCB9 | REFS | 13 | 46 | 76 | | | |
| 1 | NFFDVS | INTEGER | ARRAY | FCB9 | REFS | 13 | 42 | 44 | 68 | 71 | 75 |
| | | | | | DEFINED | 43 | 72 | | | | |
| 225 | NL | INTEGER | | | REFS | 34 | 35 | 59 | 60 | 72 | 73 |
| | | | | | 90 | 91 | DEFINED | 33 | 58 | 71 | 88 |
| 0 | NNFFDV | INTEGER | | FCB9 | REFS | 13 | 41 | 67 | | | 89 |
| 0 | STS | INTEGER | ARRAY | STATUS | REFS | 15 | 19 | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES | | | | |
|---|---|---|---|---|---|---|---|
| SETSTS | | 1 | 37 | 46 | 62 | 76 | 93 |

| STATEMENT LABELS | | DEF LINE | REFERENCES | | |
|---|---|---|---|---|---|
| 16 | 100 | 27 | 24 | | |
| 37 | 120 | 38 | 30 | 31 | 35 |
| 62 | 130 | 47 | 41 | 42 | 44 |
| 66 | 200 | 51 | 24 | | |

| STATEMENT LABELS | | DEF LINE | REFERENCES | | | | |
|---|---|---|---|---|---|---|---|
| 110 | 220 | 63 | 55 | 56 | 60 | | |
| 154 | 230 | 77 | 67 | 68 | 69 | 73 | 75 |
| 0 | 250 | 78 | 66 | | | | |
| 163 | 300 | 82 | 24 | | | | |
| 212 | 320 | 94 | 85 | 86 | 90 | 91 | |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | |
|---|---|---|---|---|---|---|
| 22 | 120 | * I | 30 38 | 20B | EXT REFS | |
| 44 | 130 | * I | 41 47 | 21B | EXT REFS | |
| 73 | 220 | * I | 55 63 | 20B | EXT REFS | |
| 120 | 250 | * I | 66 78 | 41B | EXT REFS | NOT INNER |
| 121 | 230 | * J | 67 77 | 36B | EXT REFS | |
| 167 | 320 | * I | 85 94 | 26B | EXT REFS | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | |
|---|---|---|---|---|---|
| FCB2 | 61 | 0 NBTU (1) | 1 BTUTYP (10) | 11 BTUCON (40) | |
| | | 51 BTUNO (10) | | | |
| FCB7 | 17 | 0 NDDUDV (1) | 1 DDUDVS (4) | 5 DDUDV (8) | |
| | | 13 DDUDVN (4) | | | |
| FCB8 | 37 | 0 NDFFDV (1) | 1 DFFDVS (6) | 7 DFFDV (24) | |
| | | 31 DFFDVN (6) | | | |
| FCB9 | 37 | 0 NNFFDV (1) | 1 NFFDVS (8) | 9 NFFDV (24) | |
| | | 33 NFFDVN (4) | | | |
| FCB10 | 19 | 0 NDFADV (1) | 1 DFADVS (3) | 4 DFADV (12) | |
| | | 16 DFADVN (3) | | | |
| STATUS | 20 | 0 STS (20) | | | |
| FLAGS | 5 | 0 FLG (5) | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 227B | 151 |
| COMMON LENGTH | 304B | 196 |

```
      SUBROUTINE MISCYC(BEG,END,NEXT)
C                                        THIS VERSION: 3 JULY 1974
C THE PURPOSE OF THIS SUBROUTINE IS TO DETERMINE IF A MINOR
C CYCLE           IS MISSED BECAUSE OF A RECOVERY PROCEDURE.
C BEG     BEGINNING OF THE RECOVERY PROCEDURE
C END     END OF RECOVERY PROCEDURE
C NEXT    5 IF THE RECOVERY IS TOO LONG
C BOR, DOR, EOR, DEOR HELP  IN DETERMINING IF WE ALREADY LOST TIME IN
C PREVIOUS ITERATION.
C
C ***************************************************************************
      COMMON/C027/DOR,DEOR,MISIT,TISL,MI,FRTI
      COMMON/C031/RATINT,ISYNC
      COMMON/CYC/LAUNMI
      COMMON/EXEC/MINCY,RTI,TODO,OLTIS,IDLE,SEQMAX,DOWMAX,MISITE
      INTEGER SEQ
      INTEGER SEQMAX
      REAL MINCY,IDLE
      REAL LAUNMI
      REAL LASUMI
C ***************************************************************************
C
C DETERMINE BEGINNING OF TIME SLOT AND KIND OF CYCLE (MINOR : MI=0)
      BOR=BEG
      EOR=END
      IF (ISYNC.EQ.0) GO TO 200
C SYNCHRONOUS CASE
      SEQ=0
      IF (BOR.NE.DEOR) GO TO 5
      BOR=DOR
      MISITE=MISITE-MISIT
      GO TO 60
    5 CONTINUE
      IF (BOR.NE.DOR) GO TO 6
      MISITE=MISITE-MISIT
      GO TO 60
    6 CONTINUE
      DOR=BOR
      TISL=AINT(BOR/RTI)*RTI
      IF (TISL.EQ.BOR) TISL=TISL-RTI
C NEW RTI
      FRTI=TISL+RTI
      IF (BOR-TISL-MINCY) 10,10,20
   10 MI=0
      IF (LAUNMI.LT.TISL-RTI)LASUMI=TISL-RTI
      GO TO 60
   20 CONTINUE
      IF ((OLTIS.EQ. TISL).AND.(BOR.LT.TISL+MINCY+(IDLE-TODO)))GO TO 10
      MI=1
      TISL=TISL+MINCY
   50 IF (LAUNMI.LT.TISL-MINCY)LASUMI=TISL-MINCY
   60 CONTINUE
      DEOR=EOR
      MISIT=0
C DETERMINE WHAT IS LEFT TO BE DONE IN THIS MINOR CYCLE
```

```
                IF (MI.EQ.1) GO TO 75
                IF (TISL.NE.OLTIS) TODO=IDLE-(END-BEG)
                IF (TISL.EQ.OLTIS) TODO=TODO- (END-BEG)
             75 CONTINUE
60         C DETERMINE IF RECOVERY LASTS AFTER NEW RTI (NO# GO TO 80)
                IF (EOR.LE. FRTI) GO TO 80
                MISIT=1+ INT((EOR-FRTI)/RTI)
                IF ((EOR-FRTI)/RTI.EQ.AINT((EOR-FRTI)/RTI)) MISIT =MISIT -1
                MISITE=MISITE+MISIT
65           100 CONTINUE
                SEQ=MAX0( INT((EOR-LASUMI-RTI)/RTI),MISIT)
                IA=INT(DOHMAX/RTI)
                IF (SEQ.LE.IA) GO TO 70
                MISITE=MISITE-SEQ+IA+1
70              SEQ=IA +1
             70 CONTINUE
                IF(SEQ.GT.SEQMAX) SEQMAX=SEQ
                OLTIS=RTI*AINT(EOR/RTI)
                IF (OLTIS.EQ.EOR) OLTIS=OLTIS-RTI
75              TODO=IDLE
                LAUNMI=OLTIS
                GO TO 110
             80 CONTINUE
           C IN CASE OF MINOR  DETERMINE IF RECOVERY LASTS OVER TIME SLOT
80              IF (MI.EQ.1) RETURN
                IF (TODO.GE.0.) GO TO 95
                MISITE=MISITE+1
                MISIT=1
                GO TO 100
85           95 CONTINUE
                OLTIS=TISL
           110  CONTINUE
                IF (RTI*SEQ.GT.DOHMAX)NEXT=5
                RETURN
90         C
           C ASYNCHRONOUS CASE
           200  CONTINUE
                IF (BOR.EQ.DOR) GO TO 210
           C WAS THERE AN INTERRUPT SINCE LAST RECOVERY
95              Y=DEOR+RTI*2.*RANF(0.)
                IF (Y.GE.BOR) GO TO 210
                Y=BOR
                MISIT=0
           209  CONTINUE
100             Y=Y+RTI*2.*RANF(0.)
           210  CONTINUE
                IF (Y.GT.EOR) GO TO 240
                MISITE=MISITE+1
                MISIT=MISIT+1
105             GO TO 209
           240  CONTINUE
                IF (MISIT.GT.SEQMAX) SEQMAX=MISIT
                IF (MISIT.GT. INT(DOHMAX/RTI)) NEXT=5
                DEOR=EOR
110             DOR=BOR
```

```
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | |
|---|---|---|---|---|---|
| 2 | MISCYC | 1 | 80 | 89 | 111 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BEG | REAL | F.P. | REFS | 24 | 57 | 58 | DEFINED | 1 | | |
| 257 | BOR | REAL | | REFS | 29 | 34 | 38 | 39 | 40 | 43 | 48 |
| | | | | 93 | 96 | 97 | 110 | DEFINED | 24 | 30 | |
| 1 | DEOR | REAL | C027 | REFS | 12 | 29 | 95 | DEFINED | 53 | 109 | |
| 0 | DOR | REAL | C027 | REFS | 12 | 30 | 34 | 93 | DEFINED | 38 | 110 |
| 6 | DOHMAX | REAL | EXEC | REFS | 15 | 67 | 88 | 108 | | | |
| 0 | END | REAL | F.P. | REFS | 25 | 57 | 58 | DEFINED | 1 | | |
| 260 | EOR | REAL | | REFS | 53 | 61 | 62 | 2*63 | 66 | 73 | 74 |
| | | | | 102 | 109 | DEFINED | 25 | | | | |
| 5 | FRTI | REAL | C027 | REFS | 12 | 61 | 62 | 2*63 | DEFINED | 42 | |
| 261 | IA | INTEGER | | REFS | 68 | 69 | 70 | DEFINED | 67 | | |
| 4 | IDLE | REAL | EXEC | REFS | 15 | 18 | 48 | 57 | 75 | | |
| 1 | ISYNC | INTEGER | C031 | REFS | 13 | 26 | | | | | |
| 256 | LASUMI | REAL | | REFS | 20 | 66 | DEFINED | 45 | 51 | | |
| 0 | LAUNMI | REAL | CYC | REFS | 14 | 19 | 45 | 51 | DEFINED | 76 | |
| 4 | MI | INTEGER | C027 | REFS | 12 | 56 | 80 | DEFINED | 44 | 49 | |
| 0 | MINCY | REAL | EXEC | REFS | 15 | 18 | 43 | 48 | 50 | 2*51 | |
| 2 | MISIT | INTEGER | C027 | REFS | 12 | 31 | 35 | 63 | 64 | 66 | 104 |
| | | | | 2*107 | 108 | DEFINED | 54 | 62 | 63 | 83 | 98 |
| | | | | 104 | | | | | | | |
| 7 | MISITE | INTEGER | EXEC | REFS | 15 | 31 | 35 | 64 | 69 | 82 | 103 |
| | | | | DEFINED | 31 | 35 | 64 | 69 | 82 | 103 | |
| 0 | NEXT | INTEGER | F.P. | DEFINED | 1 | 88 | 108 | | | | |
| 3 | OLTIS | REAL | EXEC | REFS | 15 | 48 | 57 | 58 | 2*74 | 76 | |
| | | | | DEFINED | 73 | 74 | 86 | | | | |
| 0 | RATINT | REAL | C031 | REFS | 13 | | | | | | |
| 1 | RTI | REAL | EXEC | REFS | 15 | 2*39 | 40 | 42 | 2*45 | 62 | 2*63 |
| | | | | 2*66 | 67 | 2*73 | 74 | 88 | 95 | 100 | 108 |
| 255 | SEQ | INTEGER | | REFS | 16 | 68 | 69 | 2*72 | 88 | | |
| | | | | DEFINED | 28 | 66 | 70 | | | | |
| 5 | SEQMAX | INTEGER | EXEC | REFS | 15 | 17 | 72 | 107 | DEFINED | 72 | 107 |
| 3 | TISL | REAL | C027 | REFS | 12 | 2*40 | 42 | 43 | 2*45 | 2*48 | 50 |
| | | | | 2*51 | 57 | 58 | 86 | DEFINED | 39 | 40 | 50 |
| 2 | TODO | REAL | EXEC | REFS | 15 | 48 | 58 | 81 | DEFINED | 57 | 58 |
| | | | | 75 | | | | | | | |
| 262 | Y | REAL | | REFS | 96 | 100 | 102 | DEFINED | 95 | 97 | 100 |

| EXTERNALS | TYPE | ARGS | REFERENCES | |
|---|---|---|---|---|
| RANF | REAL | 1 | 95 | 100 |

| INLINE FUNCTIONS | TYPE | ARGS | | DEF LINE | REFERENCES | | | |
|---|---|---|---|---|---|---|---|---|
| AINT | REAL | 1 | INTRIN | | 39 | 63 | 73 | |
| INT | INTEGER | 1 | INTRIN | | 62 | 66 | 67 | 108 |
| MAX0 | INTEGER | 0 | INTRIN | | 66 | | | |

| STATEMENT LABELS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 21 | 5 | 33 | 29 |
| 26 | 6 | 37 | 14 |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | | | |
|---|---|---|---|---|---|---|---|
| 43 | 10 | | 44 | 2*43 | 48 | | |
| 52 | 20 | | 47 | 43 | | | |
| 0 | 50 | INACTIVE | 51 | | | | |
| 70 | 60 | | 52 | 32 | 36 | 46 | |
| 144 | 70 | | 71 | 68 | | | |
| 111 | 75 | | 59 | 56 | | | |
| 162 | 80 | | 78 | 61 | | | |
| 174 | 95 | | 85 | 81 | | | |
| 126 | 100 | | 65 | 84 | | | |
| 176 | 110 | | 87 | 77 | | | |
| 205 | 200 | | 92 | 26 | | | |
| 221 | 209 | | 99 | 105 | | | |
| 226 | 210 | | 101 | 93 | 96 | | |
| 233 | 240 | | 106 | 102 | | | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| C027 | 6 | 0 DOR (1) | 1 DEOR (1) | 2 HISIT (1) |
| | | 3 TISL (1) | 4 MI (1) | 5 FRTI (1) |
| C031 | 2 | 0 RATINT (1) | 1 ISYNC (1) | |
| CYC | 1 | 0 LAUNMI (1) | | |
| EXEC | 8 | 0 MINCY (1) | 1 RTI (1) | 2 TODO (1) |
| | | 3 OLTIS (1) | 4 IDLE (1) | 5 SEQMAX (1) |
| | | 6 DOHMAX (1) | 7 HISITE (1) | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 263B | 179 |
| COMMON LENGTH | 21B | 17 |

```
      FUNCTION MSTEPD(I,XMAT,IDIM,JDIM)
      DIMENSION XMAT(IDIM,JDIM)
      U=RANF(0.0)
      DO 10 J=1,JDIM
        IF(U.LE.XMAT(I,J)) GOTO 20
   10 CONTINUE
      STOP 4006
   20 CONTINUE
      MSTEPD=J
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | MSTEPD | 1 | 10 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | I | INTEGER | | F.P. | REFS | 5 | DEFINED | 1 | | |
| 0 | IDIM | INTEGER | | F.P. | REFS | 2 | DEFINED | 1 | | |
| 53 | J | INTEGER | | | REFS | 5 | 9 | DEFINED | 4 | |
| 0 | JDIM | INTEGER | | F.P. | REFS | 2 | 4 | DEFINED | 1 | |
| 51 | MSTEPD | INTEGER | | | DEFINED | 9 | | | | |
| 52 | U | REAL | | | REFS | 5 | DEFINED | 3 | | |
| 0 | XMAT | REAL | ARRAY | F.P. | REFS | 2 | 5 | DEFINED | 1 | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| RANF | REAL | 1 | 3 |

| STATEMENT LABELS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 0 | 10 | 6 | 4 |
| 40 | 20 | 8 | 5 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | EXITS |
|---|---|---|---|---|---|---|
| 32 | 10 | * J | 4 6 | 40 | INSTACK | |

STATISTICS

PROGRAM LENGTH 678 55

```
      SUBROUTINE  NFFFLT(MDMNO,STANO,DEV)
C
C THIS ROUTINE DETERMINES THE EFFECT OF A NONDEDICATED DEVICE FAILURE
C UPON THE FLIGHT CRITICAL BUS EQUIPMENT GROUP
C
C MDMNO  IDENTIFIES MDM
C STANO  IDENTIFIES THE PILOT STATION
C DEV    IDENTIFIES THE DEVICE
C
C*************************************************************************
      COMMON/FAULT/FLTTYP
      COMMON/CO12/NTR,IFAU
      COMMON/STATUS/STS(20)
      COMMON/FCB9/NNFFDV,NFFDVS(2,4),NFFDV(3,2,4),NFFDVN(4)
      COMMON/FCB15/NNFFTF(4,2),NNFFPF(4,2),NNFFTR(4,2)
      COMMON/FCB16/NFFTFD(4,2),NFFTFL(4,2),NFFCOV(4),NFFPFD(4,2)
      COMMON/FCBUC/FCBUCF(6)
      COMMON/FLAGS/FLG(5)
      LOGICAL FLG
      INTEGER FCBUCF
      LOGICAL GREATR
      INTEGER DEV,STANO,FLTTYP,STS
      REAL NFFPFD,NFFTFD,NFFTFL
C
      NL=NFFDVS(STANO,DEV)
      IF(NL.EQ.0) RETURN
      FLG(4)=.TRUE.
      IF(MDMNO.NE.0) GOTO 400
      IFAU=IFAU+1
      GOTO(100,200),FLTTYP
C
C                 ...A TRANSIENT AFFECTING THE WHOLE UNIT
  100 CONTINUE
      NTR=NTR+1
      NNFFTF(DEV,1)=1+NNFFTF(DEV,1)
C                 ...IS THE FAULT DETECTED AND RECOVERED FROM
      IF(GREATR(NFFTFD(DEV,1))) GOTO 800
      IF(GREATR(NFFTFL(DEV,1))) RETURN
      NNFFTR(DEV,1)=1+NNFFTR(DEV,1)
      GOTO 205
C
C                 ...A PERMANENT FAULT OCCURS
  200 CONTINUE
      NNFFPF(DEV,1)=1+NNFFPF(DEV,1)
      IF(GREATR(NFFPFD(DEV,1))) GOTO 800
C                 ...ENTER  LEAKY  TRANSIENTS
  205 CONTINUE
      NFFDVS(STANO,DEV)=0
      IF(NFFDVS(3-STANO,DEV).NE.0) RETURN
C                 ...THE SYSTEM HAS FAILED
      CALL SETSTS(NFFDVN(DEV))
      RETURN
C
C                 ...THE DEVICES INTERFACE TO ONE OF THE MDMS FAILS
  400 CONTINUE
```

```
                IF(NFFDV(MDMNO,STANO,DEV).EQ.0)RETURN
                IFAU=IFAU+1
                GOTO(500,600),FLTTYP
          C
60        C                    ...TRANSIENT FAULT
          500 CONTINUE
                NTR=NTR+1
                NNFFTF(DEV,2)=NNFFTF(DEV,2)+1
                IF((NL.EQ.1).AND.(GREATR(NFFTFD(DEV,2)))) GOTO 800
65              IF(GREATR(NFFTFL(DEV,2))) RETURN
                NNFFTR(DEV,2)=NNFFTR(DEV,2)+1
                GOTO 605
          C
          C                    ...PERMANENT  FAULT
70        600 CONTINUE
                NNFFPF(DEV,2)=NNFFPF(DEV,2)+1
          605 CONTINUE
                NFFDV(MDMNO,STANO,DEV)=0
                NFFDVS(STANO,DEV)=NL-1
75              IF(NL.EQ.1) GOTO 205
                IF(NL.GE.3) RETURN
                IF(GREATR(NFFCOV(DEV))) GOTO 800
                RETURN
          C
80        C                    ...UNCOVERED OR UNDETECTED FAULT
          800 CONTINUE
                CALL SETSTS(FCBUCF(5))
                RETURN
                END
```

SYMBOLIC REFERENCE MAP

```
ENTRY POINTS      DEF LINE     REFERENCES
    2  NFFFLT          1        26        38        49        52        56        65        76        78        83

VARIABLES     SN  TYPE           RELOCATION
    0  DEV        INTEGER              F.P.        REFS        22        25      2*35        37        38      2*39      2*44
                                                      45        48        49        51        56      2*63        64        65
                                                    2*66      2*71        73        74        77   DEFINED         1
    0  FCBUCF     INTEGER    ARRAY     FCBUC       REFS        17        20        82
    0  FLG        LOGICAL    ARRAY     FLAGS       REFS        18        19   DEFINED        27
    0  FLTTYP     INTEGER              FAULT       REFS        11        22        30        58
    1  IFAU       INTEGER              C012        REFS        12        29        57   DEFINED        29        57
    0  MDMNO      INTEGER               F.P.       REFS        28        56        73   DEFINED         1
   20  NFFCOV     INTEGER    ARRAY     FCB16       REFS        16        77
   11  NFFDV      INTEGER    ARRAY     FCB9        REFS        14        56   DEFINED        73
   41  NFFDVN     INTEGER    ARRAY     FCB9        REFS        14        51
    1  NFFDVS     INTEGER    ARRAY     FCB9        REFS        14        25        49   DEFINED        48        74
   24  NFFPFD     REAL       ARRAY     FCB16       REFS        16        23        45
    0  NFFTFD     REAL       ARRAY     FCB16       REFS        16        23        37        64
   10  NFFTFL     REAL       ARRAY     FCB16       REFS        16        23        38        65
  172  NL         INTEGER                          REFS        26        64        74        75        76
                                                  DEFINED       25
    0  NNFFDV     INTEGER              FCB9        REFS        14
   10  NNFFPF     INTEGER    ARRAY     FCB15       REFS        15        44        71   DEFINED        44        71
    0  NNFFTF     INTEGER    ARRAY     FCB15       REFS        15        35        63   DEFINED        35        63
   20  NNFFTR     INTEGER    ARRAY     FCB15       REFS        15        39        66   DEFINED        39        66
    0  NTR        INTEGER              C012        REFS        12        34        62   DEFINED        34        62
    0  STANO      INTEGER               F.P.       REFS        22        25        48        49        56        73        74
                                                  DEFINED       1
    0  STS        INTEGER    ARRAY     STATUS      REFS        13        22

EXTERNALS           TYPE    ARGS       REFERENCES
           GREATR   LOGICAL    1          21        37        38        45        64        65        77
           SETSTS              1          51        82

STATEMENT LABELS              DEF LINE    REFERENCES
   26  100                        33         30
   45  200                        43         30
   53  205                        47         40        75
   72  400                        55         28
  112  500                        61         58
  134  600                        70         58
  136  605                        72         67
  161  800                        81         37        45        64        77

COMMON BLOCKS    LENGTH     MEMBERS - BIAS NAME(LENGTH)
           FAULT      1              0 FLTTYP (1)
           C012       2              0 NTR    (1)              1 IFAU   (1)
           STATUS    20              0 STS    (20)
           FCB9      37              0 NNFFDV (1)              1 NFFDVS (8)              9 NFFDV  (24)
                                    33 NFFDVN (4)
           FCB15     24              0 NNFFTF (8)              8 NNFFPF (8)             16 NNFFTR (8)
           FCB16     28              0 NFFTFD (8)              8 NFFTFL (8)             16 NFFCOV (4)
```

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS | NAME | (LENGTH) |
|---|---|---|---|---|
| | | 20 | NFFPFD | (8) |
| FCBUC | 6 | 0 | FCBUCF | (6) |
| FLAGS | 5 | 0 | FLG | (5) |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 173B | 123 |
| COMMON LENGTH | 173B | 123 |

```
      SUBROUTINE PACK(PLACE,LOC)
      INTEGER PLACE
      DIMENSION PLACE(5)
      LOC=PLACE(5)
5     IF(LOC.LE.0) LOC=0
      DO 10 I=1,4
        J=5-I
        IF(PLACE(J).LE.0)PLACE(J)=0
        LOC=OR(SHIFT(LOC,6),PLACE(J))
10   10 CONTINUE
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | PACK | 1 | 11 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 41 | I | INTEGER | | | REFS | 7 | DEFINED | 6 | | | |
| 42 | J | INTEGER | | | REFS | 2*8 | 9 | DEFINED | 7 | | |
| 0 | LOC | INTEGER | | F.P. | REFS | 5 | 9 | DEFINED | 1 | 4 | 5 | 9 |
| 8 | PLACE | INTEGER | ARRAY | F.P. | REFS | 2 | 3 | 4 | 8 | 9 | | |
| | | | | | DEFINED | 1 | 8 | | | | | |

| INLINE FUNCTIONS | TYPE | ARGS | | DEF LINE | REFERENCES |
|---|---|---|---|---|---|
| OR | NO TYPE | 2 | INTRIN | | 9 |
| SHIFT | NO TYPE | 2 | INTRIN | | 9 |

| STATEMENT LABELS | DEF LINE | REFERENCES |
|---|---|---|
| 0 10 | 10 | 6 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|
| 26 | 10 | * I | 6 10 | 118 | OPT |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 528 | 42 |

```
      SUBROUTINE PFCBCF
      COMMON/FLTHIS/NFCV(10),NSYSF(10),PFRIO,TFRIO
      COMMON/FCBPF/PFRBUS(8),PFRBTU(10,2),PFRODU(4),PFRDFF(6),
     1 PFRNFF(4,2),PFRDFA(2)
      COMMON/FCBTF/TFRBUS(8),TFRBTU(10,2),TFRODU(4),TFRDFF(6),
     1 TFRNFF(4,2),TFRDFA(2)
      COMMON/FC83/BUSTFL,BUSCOV
      COMMON/FC85/BTUCOV(10,2),BTUTFL(10,2)
      COMMON/FC812/DDUCOV(4),DDUTFL(4),DDUTFD(4,2)
      COMMON/FC814/DFFTFD(6,4),DFFTFL(6),DFFCOV(6,3)
      COMMON/FC816/NFFTFD(4,2),NFFTFL(4,2),NFFCOV(4),NFFPFD(4,2)
      COMMON/FC818/DFATFD(3),DFATFL(3),DFACOV(3)
      COMMON/FCBNM/DDUNM(4),DFFNM(6),NFFNM(4),DFANM(3)
      COMMON/FCBNM1/BUSNM(8),NBTUNM(10)
      INTEGER DDUNM,DFFNM,DFANM,BUSNM
      REAL NFFTFD,NFFTFL,NFFCOV,NFFPFD
      PRINT 2000,PFRIO,TFRIO
      PRINT 2001
      PRINT 2002,(BUSNM(I),PFRBUS(I),TFRBUS(I),I=1,8)
      PRINT 2001
      PRINT 2003,(NBTUNM(I),(PFRBTU(I,J),J=1,2),(TFRBTU(I,J),J=1,2),I=1,
     1 10)
      PRINT 2001
      PRINT 2002,(DDUNM(I),PFRDDU(I),TFRDDU(I),I=1,4)
      PRINT 2001
      PRINT 2002,(DFFNM(I),PFRDFF(I),TFRDFF(I),I=1,6)
      PRINT 2001
      PRINT2003,(NFFNM(I),(PFRNFF(I,J),J=1,2),(TFRNFF(I,J),J=1,2),I=1,4)
      PRINT 2001
      PRINT 2002,(DFANM(I),PFRDFA(I),TFRDFA(I),I=1,2)
 2000 FORMAT(1H1,47HFLIGHT CRITICAL BUS PARTITION --- FAILURE RATES ///
     1 27H NET PERMANENT FAILURE RATE,F15.2,17H  PER MILLION HRS//
     2 27H NET TRANSIENT FAILURE RATE,F15.2,17H  PER MILLION HRS///
     3 3X,6HDEVICE,15X,14HPERMANENT RATE,16X,14HTRANSIENT RATE)
 2001 FORMAT(/1X)
 2002 FORMAT(1X,A10,14X,F10.2,20X,F10.2)
 2003 FORMAT(1X,A10,10X,2F10.2,10X,2F10.2)
      PRINT 1000
      PRINT 1001,BUSTFL,BUSCOV
      PRINT 1002,(NBTUNM(I),BTUTFL(I,1),BTUTFL(I,2),BTUCOV(I,1),
     1 BTUCOV(I,2),I=1,10)
      PRINT 1003,(DDUNM(I),DDUTFD(I,1),DDUTFD(I,2),DDUTFL(I),DDUCOV(I)
     1 ,I=1,4)
      PRINT 1004,(DFFNM(I),(DFFTFD(I,J),J=1,4),DFFTFL(I),(DFFCOV(I,J)
     1 ,J=1,3),I=1,6)
      PRINT 1005,(NFFNM(I),NFFTFD(I,1),NFFTFD(I,2),NFFTFL(I,1),
     1 NFFTFL(I,2),NFFPFD(I,1),NFFCOV(I),I=1,4)
      PRINT 1006,(DFANM(I),DFATFD(I),DFATFL(I),DFACOV(I),I=1,2)
 1000 FORMAT(54H1COVERAGE PARAMETERS --- FLIGHT CRITICAL BUS PARTITION )
 1001 FORMAT(//22H BUS TRANSIENT LEAKAGE,F11.6
     1 //23H BUS PERMANENT COVERAGE,F10.6)
 1002 FORMAT(// 2X,8HBTU NAME,13X,18HTRANSIENT  LEAKAGE,12X,
     1 18HPERMANENT COVERAGE/10(/1X,A10,10X,2F10.6,10X,2F10.6))
 1003 FORMAT(// 11H ODU DEVICE,10X,20HTRANSIENT  DETECTION,10X,
     2 17HTRANSIENT LEAKAGE,13X,18HPERMANENT COVERAGE/
```

```
     3 4(/1X,A10,10X,2F10.6,10X,F14.6,16X,F14.6))
1004 FORMAT(//3X,6HDEVICE,12X,20HTRANSIENT  DETECTION,17X,
     1 17HTRANSIENT LEAKAGE,13X,19HPERMANENT  COVERAGE/
     2 6(/1X,A10,4F10.6,10X,F10.6,10X,3F10.6))
1005 FORMAT(//
     1 3X,6HDEVICE,12X,19HTRANSIENT DETECTION,13X,17HTRANSIENT LEAKAGE,
     2 11X,19HPERMANENT DETECTION,8X,18HPERMANENT COVERAGE/
     3 4(/1X,A10,10X,2F10.6,10X,2F10.6,15X,F10.6,15X,F10.6))
1006 FORMAT(//
     1 3X,6HDEVICE,8X,19HTRANSIENT DETECTION,5X,18HTRANSIENT  LEAKAGE,5X
     2 ,18HPERMANENT COVERAGE/2(/1X,A10,10X,F10.6,12X,F10.6,13X,F10.6))
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

ENTRY POINTS OFF LINE RFFERENCES
1 PFCBCF 1 67

VARIABLES SN TYPE RELOCATION

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BTUCOV | RFAL | ARRAY | FCB5 | REFS | 8 | 2*40 | | | | |
| 24 | BTUTFL | REAL | ARRAY | FCB5 | REFS | 8 | 2*40 | | | | |
| 1 | BUSCOV | REAL | | FCB3 | REFS | 7 | 39 | | | | |
| 0 | BUSNM | INTEGER | ARRAY | FCBNM1 | REFS | 14 | 15 | 19 | | | |
| 0 | BUSTFL | REAL | | FCB3 | REFS | 7 | 39 | | | | |
| 0 | DDUCOV | REAL | ARRAY | FCB12 | REFS | 9 | 42 | | | | |
| 0 | DDUNM | INTEGER | ARRAY | FCBNM | REFS | 13 | 15 | 24 | 42 | | |
| 10 | DDUTFD | REAL | ARRAY | FCB12 | REFS | 9 | 2*42 | | | | |
| 4 | DDUTFL | REAL | ARRAY | FCB12 | REFS | 9 | 42 | | | | |
| 6 | DFACOV | REAL | ARRAY | FCB18 | REFS | 12 | 48 | | | | |
| 16 | DFANM | INTEGER | ARRAY | FCBNM | REFS | 13 | 15 | 30 | 48 | | |
| 0 | DFATFD | REAL | ARRAY | FCB18 | REFS | 12 | 48 | | | | |
| 3 | DFATFL | RFAL | ARRAY | FCB18 | REFS | 12 | 48 | | | | |
| 36 | DFFCOV | REAL | ARRAY | FCB14 | REFS | 10 | 44 | | | | |
| 4 | DFFNM | INTFGER | ARRAY | FCBNM | REFS | 13 | 15 | 26 | 44 | | |
| 0 | DFFTFD | REAL | ARRAY | FCB14 | REFS | 10 | 44 | | | | |
| 30 | DFFTFL | REAL | ARRAY | FCB14 | REFS | 10 | 44 | | | | |
| 537 | I | INTEGER | | | REFS | 3*19 | 3*21 | 3*24 | 3*26 | 3*28 | 3*30 | 5*40 |
| | | | | | 5*42 | 4*44 | 7*46 | 4*48 | DEFINED | 19 | 21 | ,24 |
| | | | | | 26 | 28 | 30 | 40 | 42 | 44 | 46 | 48 |
| 540 | J | INTEGER | | | REFS | 2*21 | 2*28 | 2*44 | DEFINED | 2*21 | 2*28 | 2*44 |
| 10 | NBTUNM | INTEGER | ARRAY | FCBNM1 | REFS | 14 | 21 | 40 | | | | |
| 0 | NFCV | INTEGER | ARRAY | FLTMIS | REFS | 2 | | | | | | |
| 20 | NFFCOV | REAL | ARRAY | FCB16 | REFS | 11 | 16 | 46 | | | | |
| 12 | NFFNM | INTEGER | ARRAY | FCBNM | REFS | 13 | 28 | 46 | | | | |
| 24 | NFFPFD | REAL | ARRAY | FCB16 | REFS | 11 | 16 | 46 | | | | |
| 0 | NFFTFD | REAL | ARRAY | FCB16 | REFS | 11 | 16 | 2*46 | | | | |
| 10 | NFFTFL | REAL | ARRAY | FCB16 | REFS | 11 | 16 | 2*46 | | | | |
| 12 | NSYSF | INTEGER | ARRAY | FLTMIS | REFS | 2 | | | | | | |
| 10 | PFRBTU | REAL | ARRAY | FCBPF | REFS | 3 | 21 | | | | | |
| 0 | PFRBUS | REAL | ARRAY | FCBPF | REFS | 3 | 19 | | | | | |
| 34 | PFRDDU | REAL | ARRAY | FCBPF | REFS | 3 | 24 | | | | | |
| 56 | PFRDFA | REAL | ARRAY | FCBPF | REFS | 3 | 30 | | | | | |
| 40 | PFRDFF | REAL | ARRAY | FCBPF | REFS | 3 | 26 | | | | | |
| 24 | PFRIO | REAL | | FLTMIS | REFS | 2 | 17 | | | | | |
| 46 | PFRNFF | REAL | ARRAY | FCBPF | REFS | 3 | 28 | | | | | |
| 10 | TFRBTU | REAL | ARRAY | FCBTF | REFS | 5 | 21 | | | | | |
| 0 | TFRBUS | REAL | ARRAY | FCBTF | REFS | 5 | 19 | | | | | |
| 34 | TFRDDU | REAL | ARRAY | FCBTF | REFS | 5 | 24 | | | | | |
| 56 | TFRDFA | REAL | ARRAY | FCBTF | REFS | 5 | 30 | | | | | |
| 40 | TFRDFF | REAL | ARRAY | FCBTF | REFS | 5 | 26 | | | | | |
| 25 | TFRIO | REAL | | FLTMIS | REFS | 2 | 17 | | | | | |
| 46 | TFRNFF | REAL | ARRAY | FCBTF | REFS | 5 | 28 | | | | | |

FILE NAMES MODE
OUTPUT FMT

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| WRITES | 17 | 18 | 19 | 20 | 21 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | 38 | 39 | 40 | 42 |
| 44 | 46 | 48 | | | | | | |

| STATEMENT LABELS | | | OFF LINE | REFERENCES | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 412 | 1000 | FMT | 49 | 38 | | | | | |
| 421 | 1001 | FMT | 50 | 39 | | | | | |
| 431 | 1002 | FMT | 52 | 40 | | | | | |
| 444 | 1003 | FMT | 54 | 42 | | | | | |
| 462 | 1004 | FMT | 57 | 44 | | | | | |
| 500 | 1005 | FMT | 60 | 46 | | | | | |
| 521 | 1006 | FMT | 64 | 48 | | | | | |
| 350 | 2000 | FMT | 31 | 17 | | | | | |
| 400 | 2001 | FMT | 35 | 18 | 20 | 23 | 25 | 27 | 29 |
| 402 | 2002 | FMT | 36 | 19 | 24 | 26 | 30 | | |
| 406 | 2003 | FMT | 37 | 21 | 28 | | | | |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | |
|---|---|---|---|---|---|---|
| 17 | | * I | 19 | 118 | EXT REFS | |
| 37 | | * I | 21 | 248 | EXT REFS | NOT INNER |
| 42 | | * J | 21 | 78 | EXT REFS | |
| 52 | | * J | 21 | 78 | EXT REFS | |
| 72 | | * I | 24 | 118 | EXT REFS | |
| 112 | | * I | 26 | 118 | EXT REFS | |
| 132 | | * I | 28 | 228 | EXT REFS | NOT INNER |
| 135 | | * J | 28 | 68 | EXT REFS | |
| 144 | | * J | 28 | 68 | EXT REFS | |
| 163 | | * I | 30 | 118 | EXT REFS | |
| 212 | | * I | 40 | 158 | EXT REFS | |
| 233 | | * I | 42 | 158 | EXT REFS | |
| 254 | | * I | 44 | 268 | EXT REFS | NOT INNER |
| 257 | | * J | 44 | 78 | EXT REFS | |
| 271 | | * J | 44 | 78 | EXT REFS | |
| 306 | | * I | 46 | 218 | EXT REFS | |
| 333 | | * I | 48 | 138 | EXT REFS | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| FLTMIS | 22 | 0 NFCV (10) | 10 NSYSF (10) | 20 PFRIO (1) | | | |
| | | 21 TFRIO (1) | | | | | |
| FCBPF | 48 | 0 PFRBUS (8) | 8 PFRBTU (20) | 28 PFRDDU (4) | | | |
| | | 32 PFRDFF (6) | 38 PFRNFF (8) | 46 PFRDFA (2) | | | |
| FCBTF | 48 | 0 TFRBUS (8) | 8 TFRBTU (20) | 28 TFRDDU (4) | | | |
| | | 32 TFRDFF (6) | 38 TFRNFF (8) | 46 TFRDFA (2) | | | |
| FCB3 | 2 | 0 BUSTFL (1) | 1 BUSCOV (1) | | | | |
| FCB5 | 40 | 0 BTUCOV (20) | 20 BTUTFL (20) | | | | |
| FCB12 | 16 | 0 DDUCOV (4) | 4 DDUTFL (4) | 8 DDUTFD (8) | | | |
| FCB14 | 48 | 0 DFFTFD (24) | 24 DFFTFL (6) | 30 DFFCOV (18) | | | |
| FCB16 | 28 | 0 NFFTFD (8) | 8 NFFTFL (8) | 16 NFFCOV (4) | | | |
| | | 20 NFFPFD (8) | | | | | |
| FCB18 | 9 | 0 DFATFD (3) | 3 DFATFL (3) | 6 DFACOV (3) | | | |
| FCBNM | 17 | 0 DDUNM (4) | 4 DFFNM (6) | 10 NFFNM (4) | | | |
| | | 14 DFANM (3) | | | | | |
| FCBNM1 | 18 | 0 BUSNM (8) | 8 NBTUNM (10) | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 5418 | 353 |
| COMMON LENGTH | 4508 | 296 |

```
      SUBROUTINE PFISO(LOC,IP1)
      COMMON/RFRFCB/PF0(6),PF1(8),PF2(10),PF2X(10,4),PF3(2),PF3X(2,4)
     1,PF4(4),PF4X(4,6),PF5(4),PF5X(4,4),PF5Y(4,2),PF6(4),PF6X(4,3)
      DIMENSION IP(5)
      IGRP=ISTEPD(6,PF0)
      IP(1)=IP1
      IP(2)=IGRP
      IP(5)=0
      GOTO(1000,2000,3000,4000,5000,6000),IGRP
 1000   CONTINUE
          IP(3)=ISTEPD(8,PF1)
          IP(4)=0
          GOTO 9000
 2000   CONTINUE
          MDM=ISTEPD(10,PF2)
          IP(3)=MDM
          IP(4)=MSTEPD(MDM,PF2X,10,4)-1
          GOTO 9000
 3000   CONTINUE
          MDM=ISTEPD(2,PF3)
          IP(3)=MDM
          IP(4)=MSTEPD(MDM,PF3X,2,4)
          GOTO 9000
 4000   CONTINUE
          MDM=ISTEPD(4,PF4)
          IP(3)=MDM
          IP(4)=MSTEPD(MDM,PF4X,4,6)
          GOTO 9000
 5000   CONTINUE
          IDEV=ISTEPD(4,PF5)
          IP(5)=IDEV
          IP(3)=MSTEPD(IDEV,PF5X,4,4)-1
          IP(4)=MSTEPD(IDEV,PF5Y,4,2)
          GOTO 9000
 6000   CONTINUE
          MDM=ISTEPD(4,PF6)
          IP(3)=MDM
          IP(4)=MSTEPD(MDM,PF6X,4,3)
          GOTO 9000
 9000 CONTINUE
      CALL PACK(IP,LOC)
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | PFISO | 1 | 42 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 175 | IDEV | INTEGER | | | REFS | 31 | 32 | 33 | DEFINED | 30 | | |
| 173 | IGRP | INTEGER | | | REFS | 7 | 9 | DEFINED | 5 | | | |
| 176 | IP | INTEGER | ARRAY | | REFS | 4 | 41 | DEFINED | 6 | 7 | 8 | 11 |
| | | | | | | 12 | 16 | 17 | 21 | 22 | 26 | 27 | 31 |
| | | | | | | 32 | 33 | 37 | 38 | | | | |
| 0 | IP1 | INTEGER | | F.P. | REFS | 6 | DEFINED | 1 | | | | |
| 0 | LOC | INTEGER | | F.P. | REFS | 41 | DEFINED | 1 | | | | |
| 174 | MDM | INTEGER | | | REFS | 16 | 17 | 21 | 22 | 26 | 27 | 37 |
| | | | | | | 38 | DEFINED | 15 | 20 | 25 | 36 | |
| 0 | PF0 | REAL | ARRAY | RFRFCB | REFS | 2 | 5 | | | | | |
| 6 | PF1 | REAL | ARRAY | RFRFCB | REFS | 2 | 11 | | | | | |
| 16 | PF2 | REAL | ARRAY | RFRFCB | REFS | 2 | 15 | | | | | |
| 30 | PF2X | REAL | ARRAY | RFRFCB | REFS | 2 | 17 | | | | | |
| 100 | PF3 | REAL | ARRAY | RFRFCB | REFS | 2 | 20 | | | | | |
| 102 | PF3X | REAL | ARRAY | RFRFCB | REFS | 2 | 22 | | | | | |
| 112 | PF4 | REAL | ARRAY | RFRFCB | REFS | 2 | 25 | | | | | |
| 116 | PF4X | REAL | ARRAY | RFRFCB | REFS | 2 | 27 | | | | | |
| 146 | PF5 | REAL | ARRAY | RFRFCB | REFS | 2 | 30 | | | | | |
| 152 | PF5X | REAL | ARRAY | RFRFCB | REFS | 2 | 32 | | | | | |
| 172 | PF5Y | REAL | ARRAY | RFRFCB | REFS | 2 | 33 | | | | | |
| 202 | PF6 | REAL | ARRAY | RFRFCB | REFS | 2 | 36 | | | | | |
| 206 | PF6X | REAL | ARRAY | RFRFCB | REFS | 2 | 38 | | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ISTEPD | INTEGER | 2 | 5 | 11 | 15 | 20 | 25 | 30 | 36 |
| MSTEPD | INTEGER | 4 | 17 | 22 | 27 | 32 | 33 | 38 | |
| PACK | | 2 | 41 | | | | | | |

| STATEMENT LABELS | | DEF LINE | REFERENCES | | | | | |
|---|---|---|---|---|---|---|---|---|
| 24 | 1000 | 10 | 9 | | | | | |
| 30 | 2000 | 14 | 9 | | | | | |
| 37 | 3000 | 19 | 9 | | | | | |
| 45 | 4000 | 24 | 9 | | | | | |
| 53 | 5000 | 29 | 9 | | | | | |
| 64 | 6000 | 35 | 9 | | | | | |
| 72 | 9000 | 40 | 13 | 18 | 23 | 28 | 34 | 39 |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS | NAME | (LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| RFRFCB | 146 | 0 | PF0 | (6) | 6 | PF1 | (8) | 14 | PF2 | (10) |
| | | 24 | PF2X | (40) | 64 | PF3 | (2) | 66 | PF3X | (8) |
| | | 74 | PF4 | (4) | 78 | PF4X | (24) | 102 | PF5 | (4) |
| | | 106 | PF5X | (16) | 122 | PF5Y | (8) | 130 | PF6 | (4) |
| | | 134 | PF6X | (12) | | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 203B | 131 |
| COMMON LENGTH | 222B | 146 |

```
      SUBROUTINE PIOCNF
      COMMON/FCB1/BUSSTS(8),FCB(8,10),BTUSTS(10)
      COMMON/FCB7/NDDUDV,DDUDVS(4),DDUDV(2,4),DDUDVN(4)
      COMMON/FCB8/NDFFDV,DFFDVS(6),DFFDV(4,6),DFFDVN(6)
      COMMON/FCB9/NNFFDV,NFFDVS(2,4),NFFDV(3,2,4),NFFDVN(4)
      COMMON/FCB10/NDFADV,DFADVS(3),DFADV(4,3),DFADVN(3)
      INTEGER BTUSTS,BUSSTS,FCB,DDUDV,DDUDVN,DDUDVS,DFFDV
      INTEGER DFFDVN,DFFDVS,DFADVS,DFADV,DFADVN
      PRINT 1001,((FCB(I,J),J=1,10),I=1,8)
      PRINT 1000,BTUSTS
      PRINT 1002,((DDUDV(I,J),J=1,4),I=1,2)
      PRINT 1000,DBUDVS
      PRINT 1003,((DFFDV(I,J),J=1,6),I=1,4)
      PRINT 1000,DFFDVS
      PRINT 2004,(((NFFDV(I,J,K),J=1,2),K=1,4),I=1,3)
      PRINT 1000,NFFDVS
      PRINT 1005,((DFADV(I,J),J=1,3),I=1,4)
      PRINT 1000,DFADVS
 1000 FORMAT(10I3)
 1001 FORMAT(///8(10I3/))
 1002 FORMAT(///2(4I3/))
 1003 FORMAT(///4(6I3/))
 2004 FORMAT(///8(8I3/))
 1005 FORMAT(///4(3I3/))
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 1 | PIOCNF | 1 | 25 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 130 | BTUSTS | INTEGER | ARRAY | FCB1 | REFS | 2 | 7 | 10 | | | |
| 0 | BUSSTS | INTEGER | ARRAY | FCB1 | REFS | 2 | 7 | | | | |
| 5 | DDUDV | INTEGER | ARRAY | FCB7 | REFS | 3 | 7 | 11 | | | |
| 15 | DDUDVN | INTEGER | ARRAY | FCB7 | REFS | 3 | 7 | | | | |
| 1 | DDUDVS | INTEGER | ARRAY | FCB7 | REFS | 3 | 7 | 12 | | | |
| 4 | DFADV | INTEGER | ARRAY | FCB10 | REFS | 6 | 8 | 17 | | | |
| 20 | DFADVN | INTEGER | ARRAY | FCB10 | REFS | 6 | 8 | | | | |
| 1 | DFADVS | INTEGER | ARRAY | FCB10 | REFS | 6 | 8 | 18 | | | |
| 7 | DFFDV | INTEGER | ARRAY | FCB8 | REFS | 4 | 7 | 13 | | | |
| 37 | DFFDVN | INTEGER | ARRAY | FCB8 | REFS | 4 | 8 | | | | |
| 1 | DFFDVS | INTEGER | ARRAY | FCB8 | REFS | 4 | 8 | 14 | | | |
| 10 | FCB | INTEGER | ARRAY | FCB1 | REFS | 2 | 7 | 9 | | | |
| 164 | I | INTEGER | | | REFS | 9 | 11 | 13 | 15 | 17 | |
| | | | | | DEFINED | 9 | 11 | 13 | 15 | 17 | |
| 165 | J | INTEGER | | | REFS | 9 | 11 | 13 | 15 | 17 | |
| | | | | | DEFINED | 9 | 11 | 13 | 15 | 17 | |
| 166 | K | INTEGER | | | REFS | 15 | DEFINED | 15 | | | |
| 0 | NDDUDV | INTEGER | | FCB7 | REFS | 3 | | | | | |
| 0 | NDFADV | INTEGER | | FCB10 | REFS | 6 | | | | | |
| 0 | NDFFDV | INTEGER | | FCB8 | REFS | 4 | | | | | |
| 11 | NFFDV | INTEGER | ARRAY | FCB9 | REFS | 5 | 15 | | | | |
| 41 | NFFDVN | INTEGER | ARRAY | FCB9 | REFS | 5 | | | | | |
| 1 | NFFDVS | INTEGER | ARRAY | FCB9 | REFS | 5 | 16 | | | | |
| 0 | NNFFDV | INTEGER | | FCB9 | REFS | 5 | | | | | |

| FILE NAMES | MODE | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| OUTPUT | FMT | WRITES | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| | | | 17 | 18 | | | | | | |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | | | | |
|---|---|---|---|---|---|---|---|---|
| 143 | 1000 | FMT | 19 | 10 | 12 | 14 | 16 | 18 |
| 145 | 1001 | FMT | 20 | 9 | | | | |
| 150 | 1002 | FMT | 21 | 11 | | | | |
| 153 | 1003 | FMT | 22 | 13 | | | | |
| 161 | 1005 | FMT | 24 | 17 | | | | |
| 156 | 2004 | FMT | 23 | 15 | | | | |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | |
|---|---|---|---|---|---|---|
| 5 | | * I | 9 | 118 | EXT REFS | NOT INNER |
| 6 | | * J | 9 | 68 | EXT REFS | |
| 27 | | * I | 11 | 118 | EXT REFS | NOT INNER |
| 30 | | * J | 11 | 68 | EXT REFS | |
| 51 | | * I | 13 | 118 | EXT REFS | NOT INNER |
| 52 | | * J | 13 | 68 | EXT REFS | |
| 73 | | * I | 15 | 178 | EXT REFS | NOT INNER |
| 74 | | * K | 15 | 138 | EXT REFS | NOT INNER |
| 75 | | * J | 15 | 108 | EXT REFS | |
| 123 | | * I | 17 | 118 | EXT REFS | NOT INNER |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|
| 124 | | * J | 17 | 68 | EXT REFS |

COMMON BLOCKS

| | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FCB1 | 98 | 0 | BUSSTS | (8) | 8 | FCB | (80) | 88 | BTUSTS | (10) | |
| FCB7 | 17 | 0 | NDDUDV | (1) | 1 | DDUDVS | (4) | 5 | DDUDV | (8) | |
| | | 13 | DDUDVN | (4) | | | | | | | |
| FCB8 | 37 | 0 | NDFFDV | (1) | 1 | DFFDVS | (6) | 7 | DFFDV | (24) | |
| | | 31 | DFFDVN | (6) | | | | | | | |
| FCB9 | 37 | 0 | NNFFDV | (1) | 1 | NFFDVS | (8) | 9 | NFFDV | (24) | |
| | | 33 | NFFDVN | (4) | | | | | | | |
| FCB10 | 19 | 0 | NDFADV | (1) | 1 | DFADVS | (3) | 4 | DFADV | (12) | |
| | | 16 | DFADVN | (3) | | | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 1678 | 119 |
| COMMON LENGTH | 3208 | 208 |

```
      SUBROUTINE PIOSTS
      COMMON/FLTMIS/NFCV(10),NSYSF(10),PFRIO,TFRIO
      COMMON/FCBCNT/FCBSF,FALFCB(50)
      COMMON/FCB4/NBUSTF,NBUSPF,NBUSTR
      COMMON/C012/NTR,IFAU
      COMMON/FCB11/NDDUTF(4),NDDUPF(4),NDDUTR(4)
      COMMON/FCB13/NDFFTF(6),NDFFPF(6),NDFFTR(6)
      COMMON/FCB15/NNFFTF(4,2),NNFFPF(4,2),NNFFTR(4,2)
      COMMON/FCB17/NDFATF(3),NDFAPF(3),NDFATR(3)
      INTEGER FCBSF,FALFCB
      COMMON/FCBNM/DDUNM(4),DFFNM(6),NFFNM(4),DFANM(3)
      DIMENSION NN1(3),NN2(3),NN3(3)
      INTEGER DDUNM,DFFNM,DFANM
      COMMON/FCB6/NBTUTF(10,2),NBTUPF(10,2),NBTUTR(10,2)
      DO 10 I=1,3
        NN1(I)=NBTUTF(I,1)+NBTUTF(I,2)
        NN2(I)=NBTUPF(I,1)+NBTUPF(I,2)
      NN3(I)=NBTUTR(I,1)+NBTUTR(I,2)
   10 CONTINUE
      PRINT 1000
      NPR=IFAU-NTR
      PRINT1001,NTR,NPR,IFAU
      PRINT 1002,FCBSF
      NUNSF=FALFCB(1)+FALFCB(2)+FALFCB(3)+FALFCB(4)+FALFCB(5)+FALFCB(6)
      PRINT 1003,NUNSF
      PRINT 1004,(FALFCB(I),I=1,6)
      PRINT 1005
      PRINT 1006,NBUSTF,NBUSPF,NBUSTR,FALFCB(24)
      PRINT 1007,NN1(1),NN2(1),NN3(1),FALFCB(25)
      DO 20 I=1,4
        PRINT 1008,DDUNM(I),NDDUTF(I),NDDUPF(I),NDDUTR(I),FALFCB(I+6)
   20 CONTINUE
      PRINT 1009,NN1(2),NN2(2),NN3(2),FALFCB(26)
      DO 30 I=1,6
        PRINT 1008,DFFNM(I),NDFFTF(I),NDFFPF(I),NDFFTR(I),FALFCB(I+10)
   30 CONTINUE
      PRINT 1020
      DO 40 I=1,4
        N1=NNFFTF(I,1)+NNFFTF(I,2)
        N2=NNFFPF(I,1)+NNFFPF(I,2)
        N3=NNFFTR(I,1)+NNFFTR(I,2)
      PRINT 1008,NFFNM(I),N1,N2,N3,FALFCB(I+16)
   40 CONTINUE
      PRINT 1010,NN1(3),NN2(3),NN3(3),FALFCB(27)
      DO 50 I=1,2
      PRINT 1008,DFANM(I),NDFATF(I),NDFAPF(I),NDFATR(I),FALFCB(I+20)
   50 CONTINUE
      PRINT 1030,(I,I=1,10),(NFCV(I),I=1,10),(NSYSF(I),I=1,10)
      RETURN
 1000 FORMAT(
     1 56H1 FLIGHT  CRITICAL  BUS  PARTITION --- MISSION  STATISTICS  )
 1001 FORMAT(/
     1 120H    TRANSIENT FAULTS    ,I7/
     2 220H    PERMANENT FAULTS    ,I7/
     3 320H      TOTAL  FAULTS     ,I7/)
```

```
      1002 FORMAT(37H   NUMBER OF FLIGHT CRITICAL FAILURES  ,I7)
      1003 FORMAT(/30H   UNCOVERED SYSTEM FAILURES    ,I7)
      1004 FORMAT(/
     1    22H      BUS                ,I7/
60   2    22H      BTU                ,I7/
     3    22H      ODU DEVICE         ,I7/
     4    22H      FF-MDM DEVICE     /
     5    22H         DEDICATED      ,I7/
     6    22H         NON-DEDICATED  ,I7/
65   7    22H      FA-MDM DEVICE      ,I7//)
      1005 FORMAT(44X,23HUNCOVERED     UNCOVERED /
     1    19X,50HTRANSIENT   PERMANENT    TRANSIENT     PERMANENT   )
      1006 FORMAT(/6X,3HBUS,10X,I7,5X,I7,6X,I7,6X,I7)
      1007 FORMAT(/6X,3HODU,10X,I7,5X,I7,6X,I7,6X,I7)
70    1009 FORMAT(/6X,6HFF-MDM,7X,I7,5X,I7,6X,I7,6X,I7)
      1010 FORMAT(/6X,6HFA-MDM,7X,I7,5X,I7,6X,I7,6X,I7)
      1008 FORMAT(8X,A10,1X,I7,5X,I7,6X,I7,6X,I7)
      1020 FORMAT(1X)
      1030 FORMAT(///
75   1 26H NUMBER OF FAULTS/MISSION ,10I8//
     2 26H NUMBER OF MISSIONS       ,10I8//
     3 26H NUMBER OF SYSTEM FAILURES,10I8)
       END
```

SYMBOLIC REFERENCE MAP

ENTRY POINTS DEF LINE REFERENCES
1 PIOSTS 1 49

VARIABLES SN TYPE RELOCATION

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | DDUNM | INTEGER | ARRAY | FCBNM | REFS | 11 | 13 | 31 | | | | |
| 16 | DFANM | INTEGER | ARRAY | FCBNM | REFS | 11 | 13 | 46 | | | | |
| 4 | DFFNM | INTEGER | ARRAY | FCBNM | REFS | 11 | 13 | 35 | | | | |
| 1 | FALFCB | INTEGER | ARRAY | FCBCNT | REFS | 3 | 10 | 6*24 | 26 | 28 | 29 | 31 |
| | | | | | | 33 | 35 | 42 | 44 | 46 | | |
| 0 | FCBSF | INTEGER | | FCBCNT | REFS | 3 | 10 | 23 | | | | |
| 427 | I | INTEGER | | | REFS | 3*16 | 3*17 | 3*18 | 26 | 5*31 | 5*35 | 2*39 |
| | | | | | | 2*40 | 2*41 | 2*42 | 5*46 | 3*48 | DEFINED | 15 | 26 |
| | | | | | | 30 | 34 | 38 | 45 | 3*48 | | | |
| 1 | IFAU | INTEGER | | C012 | REFS | 5 | 21 | 22 | | | | |
| 24 | NBTUPF | INTEGER | ARRAY | FCB6 | REFS | 14 | 2*17 | | | | | |
| 0 | NBTUTF | INTEGER | ARRAY | FCB6 | REFS | 14 | 2*16 | | | | | |
| 50 | NBTUTR | INTEGER | ARRAY | FCB6 | REFS | 14 | 2*18 | | | | | |
| 1 | NBUSPF | INTEGER | | FCB4 | REFS | 4 | 28 | | | | | |
| 0 | NBUSTF | INTEGER | | FCB4 | REFS | 4 | 28 | | | | | |
| 2 | NBUSTR | INTEGER | | FCB4 | REFS | 4 | 28 | | | | | |
| 4 | NDDUPF | INTEGER | ARRAY | FCB11 | REFS | 6 | 31 | | | | | |
| 0 | NDDUTF | INTEGER | ARRAY | FCB11 | REFS | 6 | 31 | | | | | |
| 10 | NDDUTR | INTEGER | ARRAY | FCB11 | REFS | 6 | 31 | | | | | |
| 3 | NDFAPF | INTEGER | ARRAY | FCB17 | REFS | 9 | 46 | | | | | |
| 0 | NDFATF | INTEGER | ARRAY | FCB17 | REFS | 9 | 46 | | | | | |
| 6 | NDFATR | INTEGER | ARRAY | FCB17 | REFS | 9 | 46 | | | | | |
| 6 | NDFFPF | INTEGER | ARRAY | FCB13 | REFS | 7 | 35 | | | | | |
| 0 | NDFFTF | INTEGER | ARRAY | FCB13 | REFS | 7 | 35 | | | | | |
| 14 | NDFFTR | INTEGER | ARRAY | FCB13 | REFS | 7 | 35 | | | | | |
| 0 | NFCV | INTEGER | ARRAY | FLTMIS | REFS | 2 | 48 | | | | | |
| 12 | NFFNM | INTEGER | ARRAY | FCBNM | REFS | 11 | 42 | | | | | |
| 10 | NNFFPF | INTEGER | ARRAY | FCB15 | REFS | 8 | 2*40 | | | | | |
| 0 | NNFFTF | INTEGER | ARRAY | FCB15 | REFS | 8 | 2*39 | | | | | |
| 20 | NNFFTR | INTEGER | ARRAY | FCB15 | REFS | 8 | 2*41 | | | | | |
| 435 | NN1 | INTEGER | ARRAY | | REFS | 12 | 29 | 33 | 44 | DEFINED | 16 | |
| 440 | NN2 | INTEGER | ARRAY | | REFS | 12 | 29 | 33 | 44 | DEFINED | 17 | |
| 443 | NN3 | INTEGER | ARRAY | | REFS | 12 | 29 | 33 | 44 | DEFINED | 18 | |
| 430 | NPR | INTEGER | | | REFS | 22 | DEFINED | 21 | | | | |
| 12 | NSYSF | INTEGER | ARRAY | FLTMIS | REFS | 2 | 48 | | | | | |
| 0 | NTR | INTEGER | | C012 | REFS | 5 | 21 | 22 | | | | |
| 431 | NUNSF | INTEGER | | | REFS | 25 | DEFINED | 24 | | | | |
| 432 | N1 | INTEGER | | | REFS | 42 | DEFINED | 39 | | | | |
| 433 | N2 | INTEGER | | | REFS | 42 | DEFINED | 40 | | | | |
| 434 | N3 | INTEGER | | | REFS | 42 | DEFINED | 41 | | | | |
| 24 | PFRIO | REAL | | FLTMIS | REFS | 2 | | | | | | |
| 25 | TFRIO | REAL | | FLTMIS | REFS | 2 | | | | | | |

FILE NAMES MODE

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| OUTPUT | FMT | WRITES | 20 | 22 | 23 | 25 | 26 | 27 | 28 | 29 |
| | | 31 | 33 | 35 | 37 | 42 | 44 | 46 | 48 | |

| STATEMENT LABELS | | | DEF LINE | REFERENCES | | | |
|---|---|---|---|---|---|---|---|
| 0 | 10 | | 19 | 15 | | | |
| 0 | 20 | | 32 | 30 | | | |
| 0 | 30 | | 36 | 34 | | | |
| 0 | 40 | | 43 | 38 | | | |
| 0 | 50 | | 47 | 45 | | | |
| 267 | 1000 | FMT | 50 | 20 | | | |
| 277 | 1001 | FMT | 52 | 22 | | | |
| 311 | 1002 | FMT | 56 | 23 | | | |
| 317 | 1003 | FMT | 57 | 25 | | | |
| 324 | 1004 | FMT | 58 | 26 | | | |
| 351 | 1005 | FMT | 66 | 27 | | | |
| 363 | 1006 | FMT | 68 | 28 | | | |
| 367 | 1007 | FMT | 69 | 29 | | | |
| 405 | 1008 | FMT | 72 | 31 | 35 | 42 | 46 |
| 373 | 1009 | FMT | 70 | 33 | | | |
| 400 | 1010 | FMT | 71 | 44 | | | |
| 411 | 1020 | FMT | 73 | 37 | | | |
| 413 | 1030 | FMT | 74 | 48 | | | |

| LOOPS | LABEL | | INDEX | FROM-TO | LENGTH | PROPERTIES | |
|---|---|---|---|---|---|---|---|
| 6 | 10 | | I | 15 19 | 60 | INSTACK | |
| 110 | 20 | * | I | 30 32 | 200 | | EXT REFS |
| 144 | 30 | * | I | 34 36 | 200 | | EXT REFS |
| 170 | 40 | * | I | 38 43 | 260 | | EXT REFS |
| 232 | 50 | * | I | 45 47 | 200 | | EXT REFS |
| 255 | | * | I | 48 | 50 | | EXT REFS |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| FLTMIS | 22 | 0 | NFCV | (10) | 10 | NSYSF (10) | 20 PFRIO (1) |
| | | 21 | TFRIO | (1) | | | |
| FCBCNT | 51 | 0 | FCBSF | (1) | 1 | FALFCB (50) | |
| FCB4 | 3 | 0 | NBUSTF | (1) | 1 | NBUSPF (1) | 2 NBUSTR (1) |
| C012 | 2 | 0 | NTR | (1) | 1 | IFAU (1) | |
| FCB11 | 12 | 0 | NDDUTF | (4) | 4 | NDDUPF (4) | 8 NDDUTR (4) |
| FCB13 | 18 | 0 | NDFFTF | (6) | 6 | NDFFPF (6) | 12 NDFFTR (6) |
| FCB15 | 24 | 0 | NNFFTF | (8) | 8 | NNFFPF (8) | 16 NNFFTR (8) |
| FCB17 | 9 | 0 | NDFATF | (3) | 3 | NDFAPF (3) | 6 NDFATR (3) |
| FCBNM | 17 | 0 | DDUNM | (4) | 4 | DFFNM (6) | 10 NFFNM (4) |
| | | 14 | DFANM | (3) | | | |
| FCB6 | 60 | 0 | NBTUTF | (20) | 20 | NBTUPF (20) | 40 NBTUTR (20) |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 4468 | 294 |
| COMMON LENGTH | 3328 | 218 |

```
      SUBROUTINE RDIOFR(PFRIO,TFRIO)
      COMMON/RFRFCB/PF0(6),PF1(8),PF2(10),PF2X(10,4),PF3(2),PF3X(2,4)
     1,PF4(4),PF4X(4,6),PF5(4),PF5X(4,4),PF5Y(4,2),PF6(4),PF6X(4,3)
      COMMON/TFRFCB/TF0(6),TF1(8),TF2(10),TF2X(10,4),TF3(2),TF3X(2,4)
     1,TF4(4),TF4X(4,6),TF5(4),TF5X(4,4),TF5Y(4,2),TF6(4),TF6X(4,3)
      COMMON/IFCB7/NODUDV,DDUDVS(4),DDUDV(2,4)
      COMMON/IFCB8/NDFFDV,DFFDVS(6),DFFDV(4,6)
      COMMON/IFCB9/NNFFDV,NFFDVS(2,4),NFFDV(3,2,4)
      COMMON/IFCB10/NDFADV,DFADVS(3),DFADV(4,3)
      COMMON/IFCB1/BUSSTS(8),FCB(8,10),BTUSTS(10)
      COMMON/FCBPF/PFRBUS(8),PFRBTU(10,2),PFRDDU(4),PFRDFF(6),
     1 PFRNFF(4,2),PFRDFA(2)
      COMMON/FCBTF/TFRBUS(8),TFRBTU(10,2),TFRDDU(4),TFRDFF(6),
     1 TFRNFF(4,2),TFRDFA(2)
      INTEGER DDUDVS,DDUDV,DFFDVS,DFFDV,DFADVS,DFADV,BUSSTS,FCB,BTUSTS
      CALL GIORF1(TOT,PF1,PFRBUS)
      PF0(1)=TOT
      CALL GIORF2(TOT,PF2,PF2X,BTUSTS,PFRBTU)
      PF0(2)=TOT
      CALL GIORF3(TOT,PF3,PF3X,DDUDV,2,4,PFRDDU)
      PF0(3)=TOT
      CALL GIORF3(TOT,PF4,PF4X,DFFDV,4,6,PFRDFF)
      PF0(4)=TOT
      CALL GIORF4(TOT,PF5,PF5X,PF5Y,NFFDVS,PFRNFF)
      PF0(5)=TOT
      CALL GIORF3(TOT,PF6,PF6X,DFADV,4,2,PFRDFA)
      PF0(6)=TOT
      CALL GIORF1(TOT,TF1,TFRBUS)
      TF0(1)=TOT
      CALL GIORF2(TOT,TF2,TF2X,BTUSTS,TFRBTU)
      TF0(2)=TOT
      CALL GIORF3(TOT,TF3,TF3X,DDUDV,2,4,TFRDDU)
      TF0(3)=TOT
      CALL GIORF3(TOT,TF4,TF4X,DFFDV,4,6,TFRDFF)
      TF0(4)=TOT
      CALL GIORF4(TOT,TF5,TF5X,TF5Y,NFFDVS,TFRNFF)
      TF0(5)=TOT
      CALL GIORF3(TOT,TF6,TF6X,DFADV,4,2,TFRDFA)
      TF0(6)=TOT
      DO 10 I=2,6
         TF0(I)=TF0(I)+TF0(I-1)
         PF0(I)=PF0(I)+PF0(I-1)
   10 CONTINUE
      PFRIO=PF0(6)
      TFRIO=TF0(6)
      DO 20 I=1,6
         IF(PFRIO.NE.0) PF0(I)=PF0(I)/PFRIO
         IF(TFRIO.NE.0) TF0(I)=TF0(I)/TFRIO
   20 CONTINUE
      RETURN
      END
```

SYMBOLIC REFFRENCE MAP

ENTRY POINTS OFF LINE REFERENCES
2 RDIOFR 1 50

VARIABLES SN TYPE RELOCATION

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 130 | BTUSTS | INTEGER | ARRAY | IFCB1 | REFS | 10 | 15 | 18 | 30 | | | |
| 0 | BUSSTS | INTEGER | ARRAY | IFCB1 | REFS | 10 | 15 | | | | | |
| 5 | DDUDV | INTEGER | ARRAY | IFCB7 | REFS | 6 | 15 | 20 | 32 | | | |
| 1 | DDUDVS | INTEGER | ARRAY | IFCB7 | REFS | 6 | 15 | | | | | |
| 4 | DFADV | INTEGER | ARRAY | IFCB10 | REFS | 9 | 15 | 26 | 38 | | | |
| 1 | DFADVS | INTEGER | ARRAY | IFCB10 | REFS | 9 | 15 | | | | | |
| 7 | DFFDV | INTEGER | ARRAY | IFCB8 | REFS | 7 | 15 | 22 | 34 | | | |
| 1 | DFFDVS | INTEGER | ARRAY | IFCB8 | REFS | 7 | 15 | | | | | |
| 10 | FCB | INTEGER | ARRAY | IFCB1 | REFS | 10 | 15 | | | | | |
| 226 | I | INTEGER | | | REFS | 3*41 | 3*42 | 2*47 | 2*48 | DEFINED | 40 | 46 |
| 0 | NDDUDV | INTEGER | | IFCB7 | REFS | 6 | | | | | | |
| 0 | NDFADV | INTEGER | | IFCB10 | REFS | 9 | | | | | | |
| 0 | NDFFDV | INTEGER | | IFCB8 | REFS | 7 | | | | | | |
| 11 | NFFDV | INTEGER | ARRAY | IFCB9 | REFS | 8 | | | | | | |
| 1 | NFFDVS | INTEGER | ARRAY | IFCB9 | REFS | 8 | 24 | 36 | | | | |
| 0 | NNFFDV | INTEGER | | IFCB9 | REFS | 8 | | | | | | |
| 10 | PFRBTU | REAL | ARRAY | FCBPF | REFS | 11 | 18 | | | | | |
| 0 | PFRBUS | REAL | ARRAY | FCBPF | REFS | 11 | 16 | | | | | |
| 34 | PFRDDU | REAL | ARRAY | FCBPF | REFS | 11 | 20 | | | | | |
| 56 | PFRDFA | REAL | ARRAY | FCBPF | REFS | 11 | 26 | | | | | |
| 40 | PFRDFF | REAL | ARRAY | FCBPF | REFS | 11 | 22 | | | | | |
| 0 | PFRIO | REAL | | F.P. | REFS | 2*47 | DEFINED | 1 | 44 | | | |
| 46 | PFRNFF | REAL | ARRAY | FCBPF | REFS | 11 | 24 | | | | | |
| 0 | PF0 | REAL | ARRAY | RFRFCB | REFS | 2 | 2*42 | 44 | 47 | DEFINED | 17 | 19 |
| | | | | | | 21 | 23 | 25 | 27 | 42 | 47 | |
| 6 | PF1 | REAL | ARRAY | RFRFCB | REFS | 2 | 16 | | | | | |
| 16 | PF2 | REAL | ARRAY | RFRFCB | REFS | 2 | 18 | | | | | |
| 30 | PF2X | REAL | ARRAY | RFRFCB | REFS | 2 | 18 | | | | | |
| 100 | PF3 | REAL | ARRAY | RFRFCB | REFS | 2 | 20 | | | | | |
| 102 | PF3X | REAL | ARRAY | RFRFCB | REFS | 2 | 20 | | | | | |
| 112 | PF4 | REAL | ARRAY | RFRFCB | REFS | 2 | 22 | | | | | |
| 116 | PF4X | REAL | ARRAY | RFRFCB | REFS | 2 | 22 | | | | | |
| 146 | PF5 | REAL | ARRAY | RFRFCB | REFS | 2 | 24 | | | | | |
| 152 | PF5X | REAL | ARRAY | RFRFCB | REFS | 2 | 24 | | | | | |
| 172 | PF5Y | REAL | ARRAY | RFRFCB | REFS | 2 | 24 | | | | | |
| 202 | PF6 | REAL | ARRAY | RFRFCB | REFS | 2 | 26 | | | | | |
| 206 | PF6X | REAL | ARRAY | RFRFCB | REFS | 2 | 26 | | | | | |
| 10 | TFRBTU | REAL | ARRAY | FCBTF | REFS | 13 | 30 | | | | | |
| 0 | TFRBUS | REAL | ARRAY | FCBTF | REFS | 13 | 28 | | | | | |
| 34 | TFRDDU | REAL | ARRAY | FCBTF | REFS | 13 | 32 | | | | | |
| 56 | TFRDFA | REAL | ARRAY | FCBTF | REFS | 13 | 38 | | | | | |
| 40 | TFRDFF | REAL | ARRAY | FCBTF | REFS | 13 | 34 | | | | | |
| 0 | TFRIO | REAL | | F.P. | REFS | 2*48 | DEFINED | 1 | 45 | | | |
| 46 | TFRNFF | REAL | ARRAY | FCBTF | REFS | 13 | 36 | | | | | |
| 0 | TF0 | REAL | ARRAY | TFRFCB | REFS | 4 | 2*41 | 45 | 48 | DEFINED | 29 | 31 |
| | | | | | | 33 | 35 | 37 | 39 | 41 | 48 | |
| 6 | TF1 | REAL | ARRAY | TFRFCB | REFS | 4 | 28 | | | | | |
| 16 | TF2 | REAL | ARRAY | TFRFCB | REFS | 4 | 30 | | | | | |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 30 | TF2X | REAL | ARRAY | TFRFCB | REFS | 4 | 30 | | | |
| 100 | TF3 | REAL | ARRAY | TFRFCB | REFS | 4 | 32 | | | |
| 102 | TF3X | REAL | ARRAY | TFRFCB | REFS | 4 | 32 | | | |
| 112 | TF4 | REAL | ARRAY | TFRFCB | REFS | 4 | 34 | | | |
| 116 | TF4X | REAL | ARRAY | TFRFCB | REFS | 4 | 34 | | | |
| 146 | TF5 | REAL | ARRAY | TFRFCB | REFS | 4 | 36 | | | |
| 152 | TF5X | REAL | ARRAY | TFRFCB | REFS | 4 | 36 | | | |
| 172 | TF5Y | REAL | ARRAY | TFRFCB | REFS | 4 | 36 | | | |
| 202 | TF6 | REAL | ARRAY | TFRFCB | REFS | 4 | 38 | | | |
| 206 | TF6X | REAL | ARRAY | TFRFCB | REFS | 4 | 38 | | | |
| 225 | TOT | REAL | | | REFS | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| | | | | | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| | | | | | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 |
| | | | | | 39 | | | | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES | | | | | |
|---|---|---|---|---|---|---|---|---|
| GIORF1 | | 3 | 16 | 28 | | | | |
| GIORF2 | | 5 | 18 | 30 | | | | |
| GIORF3 | | 7 | 20 | 22 | 26 | 32 | 34 | 38 |
| GIORF4 | | 6 | 24 | 36 | | | | |

| STATEMENT LABELS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 0 | 10 | 43 | 40 |
| 0 | 20 | 49 | 46 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|
| 53 | 10 | I | 40 43 | 48 | INSTACK |
| 64 | 20 | I | 46 49 | 118 | OPT |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| RFRFCB | 146 | 0 PF0 (6) | 6 PF1 (8) | 14 PF2 (10) | | | |
| | | 24 PF2X (40) | 64 PF3 (2) | 66 PF3X (8) | | | |
| | | 74 PF4 (4) | 78 PF4X (24) | 102 PF5 (4) | | | |
| | | 106 PF5X (16) | 122 PF5Y (8) | 130 PF6 (4) | | | |
| | | 134 PF6X (12) | | | | | |
| TFRFCB | 146 | 0 TF0 (6) | 6 TF1 (8) | 14 TF2 (10) | | | |
| | | 24 TF2X (40) | 64 TF3 (2) | 66 TF3X (8) | | | |
| | | 74 TF4 (4) | 78 TF4X (24) | 102 TF5 (4) | | | |
| | | 106 TF5X (16) | 122 TF5Y (8) | 130 TF6 (4) | | | |
| | | 134 TF6X (12) | | | | | |
| IFCB7 | 13 | 0 NDDUDV (1) | 1 DDUDVS (4) | 5 DDUDV (8) | | | |
| IFCB8 | 31 | 0 NDFFDV (1) | 1 DFFDVS (6) | 7 DFFDV (24) | | | |
| IFCB9 | 33 | 0 NNFFDV (1) | 1 NFFDVS (8) | 9 NFFDV (24) | | | |
| IFCB10 | 16 | 0 NDFADV (1) | 1 DFADVS (3) | 4 DFADV (12) | | | |
| IFCB1 | 98 | 0 BUSSTS (8) | 8 FCB (80) | 88 BTUSTS (10) | | | |
| FCBPF | 48 | 0 PFRBUS (8) | 8 PFRBTU (20) | 28 PFRDDU (4) | | | |
| | | 32 PFRDFF (6) | 38 PFRNFF (8) | 46 PFRDFA (2) | | | |
| FCBTF | 48 | 0 TFRBUS (8) | 8 TFRBTU (20) | 28 TFRDDU (4) | | | |
| | | 32 TFRDFF (6) | 38 TFRNFF (8) | 46 TFRDFA (2) | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 2278 | 151 |
| COMMON LENGTH | 11038 | 579 |

```
      SUBROUTINE SETSTS(ISTS)
      COMMON/STATUS/STS(20)
      INTEGER STS
      N=STS(1)
5     IF(N.GT.18) STOP 1001
      STS(1)=N+1
      STS(N+2)=ISTS
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | SETSTS | 1 | 8 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ISTS | INTEGER | | F.P. | REFS | 7 | DEFINED | 1 | | | | |
| 15 | N | INTEGER | | | REFS | 5 | 6 | 7 | DEFINED | 4 | | |
| 0 | STS | INTEGER | ARRAY | STATUS | REFS | 2 | 3 | 4 | DEFINED | 6 | 7 | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) |
|---|---|---|
| STATUS | 20 | 0 STS (20) |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 20B | 16 |
| COMMON LENGTH | 24B | 20 |

```
      SUBROUTINE STATE1(NEXT)
C                        THIS VERSION:  MARCH 1976
C ACFAU(10,6)IS THE TABLE OF THE ACTIVE FAULTS. THEY ARE NO MORE THAN 10
C ACFAU(I,1) : OCCURRENCE TIME.
C ACFAU(I,2) : DISAPPEARANCE TIME.
C ACFAU(I,3) : MODULE (=0 TO INDICATE NO FAULTS).
C ACFAU(I,4) : COMPUTER.
C ACFAU(I,5) : EXTENT.
C ACFAU(I,6) : DETECTION TIME.
C DETEC   DETECTION TIME COMPUTED BY DETTIME
C EXTEN IS THE EXTENT OF THE FAULT
C IDETECT POINTS IN ACFAU TO THE NEXT FAULT TO BE DETECTED
C PTR POINTS TO THE NEXT FAULT TO OCCUR
C     NEXT  = NEXT STATE
C********** ********* ********* ********** ********* ********* ********
C
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/COM7/REASON
      COMMON/COM8/FOCO
      COMMON/CO11/MISTAK
      COMMON/CO15/NOON(5),NHORK
      COMMON/CO31/RATINT,ISYNC
      COMMON/CO38/IOCU(5),NONDED,NHOIO
      COMMON/CO42/MISTKI(3)
      DIMENSION IGO(5)
      INTEGER EXTEN,PTR
      INTEGER REASON
C ******************************************************************************
C
      IRA=0
C SUPPRESSION OF THE FAULTS IN THE SWITCHED OFF COMPUTER
      DO 10 J=1,IDIM
      IF(ACFAU(J,3).EQ.0.) GO TO 15
      I=ACFAU(J,4)
      IF (NOON(I).EQ.1) GO TO 10
      ACFAU(J,3)=0.
      IRA=IRA+1
      IF((ACFAU(J,2).GE.ENDMIS).OR.(IRA.NE.1))GOTO 10
      MISTAK=MISTAK+1
      MISTKI(NHORK-1)=MISTKI(NHORK-1)+1
   10 CONTINUE
   15 CONTINUE
      IF (IRA.GE.1) CALL GATHER
C IF THERE IS A LURKING FAULT GO TO 1
      IF (ACFAU(1,3).NE.0.) GO TO 1
    4 CONTINUE
C IF NO MORE FAULT RETURN (NEXT=6)
      TIME=TABLE(PTR,1)
      IF (TIME.LT.ENDMIS) GO TO 2
   22 CONTINUE
      NEXT=6
      RETURN
    2 CONTINUE
      CALL FIFAU(IN,NEXT,ISYNC)
```

```
      IF(NEXT.EQ.5) RETURN
      IF(IN.EQ.0) GOTO 4
C
C DETERMINE NEXT DETECTION TIME.
60     1 CONTINUE
      DET=1000000000000.                                          SP15APR4
      DO 20 I=1,IDIM
C        FIRST RECORD THATS NON-ACTIVE SAYS YOU VE LOOKED ENOUGH   SP08APR4
      IF (ACFAU(I,3).EQ.0.) GO TO 21
65    IF (ACFAU(I,6).GE. DET) GO TO 20
      DET=ACFAU(I,6)
      IDETEC=I
      FOCO=ACFAU(IDETEC,4)
   20 CONTINUE
70 ·21 CONTINUE
C
C TESTS ON MULTIPLE FAULT.
C IF NO FAULT BEFORE DET, START RECOVERY.
      TIME=AMIN1(TABLE(PTR,1),DET)
75    IF (TIME.GE.ENDMIS) GO TO 22
      IF (TIME.NE.DET) GO TO 27
      NEXT=2
C        TEST ON MULTIPLE FAULT                                    OL29APR4
      K = IDETEC + 1                                              OL29APR4
80    C IGO INDICATES A FAULT IN A COMPUTER IF 1
      IF (K.GT.IDIM) RETURN
      DO 26 I=1,5
   26 IGO(I)=0
      IGO(ACFAU(IDETEC,4))=1
85    MUL=1
      DO 23 I=K,IDIM                                              OL29APR4
      IF (ACFAU(I,3).EQ.0.)  GO TO 24                             OL29APR4
      IF (ACFAU(I,6).GT.DET) GO TO 23
      IF (IGO(ACFAU(I,4)).EQ.1) GO TO 23
90    MUL=MUL+1
      IGO(ACFAU(I,4))=1
   23 CONTINUE                                                    OL29APR4
   24 CONTINUE                                                    OL29APR4
      IF ((MUL.GE.NWORK-1).AND.( NWORK.GE.3)) NEXT=3
95    RETURN                                                      OL29APR4
C
C
C NEW FAULT BEFORE DETECTION
   27 CONTINUE
100   CALL FIFAU(IN,NEXT,ISYNC)
      IF(NEXT.EQ.5)RETURN
      IF(IN.EQ.0) GOTO 21
      GOTO 1
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | DEF LINE | REFERENCES | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 STATE1 | 1 | 53 | 56 | 81 | 95 | 101 | | | | | | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 ACFAU | | REAL | ARRAY | COM3 | REFS | 18 | 34 | 35 | 39 | 46 | 64 | 65 |
| | | | | | 66 | 68 | 84 | 87 | 88 | 89 | 91 | |
| | | | | | DEFINED | 37 | | | | | | |
| 2265 DELAY | | REAL | | COM1 | REFS | 17 | | | | | | |
| 206 DET | | REAL | | | REFS | 65 | 74 | 76 | 88 | DEFINED | 61 | 66 |
| 74 ENDMIS | | REAL | | COM3 | REFS | 18 | 39 | 50 | 75 | | | |
| 2262 EXTEN | | INTEGER | | COM1 | REFS | 17 | 27 | | | | | |
| 0 FOCO | | REAL | | COM8 | REFS | 20 | DEFINED | 68 | | | | |
| 204 I | | INTEGER | | | REFS | 36 | 64 | 65 | 66 | 67 | 83 | 87 |
| | | | | | 88 | 89 | 91 | DEFINED | 35 | 62 | 82 | 86 |
| 2263 IDETEC | | INTEGER | | COM1 | REFS | 17 | 68 | 79 | 84 | DEFINED | 67 | |
| 0 IDIM | | INTEGER | | COM1 | REFS | 17 | 33 | 62 | 81 | 86 | | |
| 211 IGO | | INTEGER | ARRAY | | REFS | 26 | 89 | DEFINED | 83 | 84 | 91 | |
| 205 IN | | INTEGER | | | REFS | 55 | 57 | 100 | 102 | | | |
| 0 IOCU | | INTEGER | ARRAY | C038 | REFS | 24 | | | | | | |
| 202 IRA | | INTEGER | | | REFS | 38 | 39 | 44 | DEFINED | 31 | 38 | |
| 1 ISYNC | | INTEGER | | C031 | REFS | 23 | 55 | 100 | | | | |
| 203 J | | INTEGER | | | REFS | 34 | 35 | 37 | 39 | DEFINED | 33 | |
| 207 K | | INTEGER | | | REFS | 81 | 86 | DEFINED | 79 | | | |
| 75 MEMSIZ | | INTEGER | | COM3 | REFS | 18 | | | | | | |
| 0 MISTAK | | INTEGER | | C011 | REFS | 21 | 40 | DEFINED | 40 | | | |
| 0 MISTKI | | INTEGER | ARRAY | C042 | REFS | 25 | 41 | DEFINED | 41 | | | |
| 210 MUL | | INTEGER | | | REFS | 90 | 94 | DEFINED | 85 | 90 | | |
| 0 NEXT | | INTEGER | | F.P. | REFS | 55 | 56 | 100 | 101 | DEFINED | 1 | 52 |
| | | | | | 77 | 94 | | | | | | |
| 5 NONDEO | | INTEGER | | C038 | REFS | 24 | | | | | | |
| 0 NOON | | INTEGER | ARRAY | C015 | REFS | 22 | 36 | | | | | |
| 6 NWOIO | | INTEGER | | C038 | REFS | 24 | | | | | | |
| 5 NWORK | | INTEGER | | C015 | REFS | 22 | 2*41 | 2*94 | | | | |
| 2261 PTR | | INTEGER | | COM1 | REFS | 17 | 27 | 49 | 74 | | | |
| 0 RATINT | | REAL | | C031 | REFS | 23 | | | | | | |
| 0 REASON | | INTEGER | | COM7 | REFS | 19 | 28 | | | | | |
| 2264 RECOV | | REAL | | COM1 | REFS | 17 | | | | | | |
| 1 TABLE | | REAL | ARRAY | COM1 | REFS | 17 | 49 | 74 | | | | |
| 76 TC | | REAL | | COM3 | REFS | 18 | | | | | | |
| 2266 TIME | | REAL | | COM1 | REFS | 17 | 50 | 75 | 76 | DEFINED | 49 | 74 |

| EXTERNALS | TYPE | ARGS | REFERENCES | |
|---|---|---|---|---|
| FIFAU | - | 3 | 55 | 100 |
| GATHER | | 0 | 44 | |

| INLINE FUNCTIONS | TYPE | ARGS | | DEF LINE | REFERENCES |
|---|---|---|---|---|---|
| AMIN1 | REAL | 0 | INTRIN | | 74 |

| STATEMENT LABELS | DEF LINE | REFERENCES | |
|---|---|---|---|
| 62 1 | 60 | 46 | 103 |
| 47 2 | 54 | 50 | |
| 40 4 | 47 | 57 | |

| STATEMENT LABELS | | DEF LINE | REFERENCES | | |
|---|---|---|---|---|---|
| 27 | 10 | 42 | 33 | 36 | 39 |
| 31 | 15 | 43 | 34 | | |
| 100 | 20 | 69 | 62 | 65 | |
| 102 | 21 | 70 | 64 | 102 | |
| 44 | 22 | 51 | 75 | | |
| 147 | 23 | 92 | 86 | 88 | 89 |
| 151 | 24 | 93 | 87 | | |
| 0 | 26 | 83 | 82 | | |
| 161 | 27 | 99 | 76 | | |

| LOOPS | LABEL | | INDEX | FROM-TO | LENGTH | PROPERTIES | |
|---|---|---|---|---|---|---|---|
| 12 | 10 | * | J | 33 42 | 17B | OPT | EXITS |
| 71 | 20 | * | I | 62 69 | 11B | OPT | EXITS |
| 123 | 26 | | I | 82 83 | 2B | INSTACK | |
| 135 | 23 | * | I | 86 92 | 14B | OPT | EXITS |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 | IDIM | (1) | 1 | TABLE | (1200) | 1201 | PTR | (1) |
| | | 1202 | EXTEN | (1) | 1203 | IDETEC | (1) | 1204 | RECOV | (1) |
| | | 1205 | DELAY | (1) | 1206 | TIME | (1) | | | |
| COM3 | 63 | 0 | ACFAU | (60) | 60 | ENDMIS | (1) | 61 | MEMSIZ | (1) |
| | | 62 | TC | (1) | | | | | | |
| COM7 | 1 | 0 | REASON | (1) | | | | | | |
| COM8 | 1 | 0 | FOCO | (1) | | | | | | |
| CO11 | 1 | 0 | MISTAK | (1) | | | | | | |
| CO15 | 6 | 0 | NOON | (5) | 5 | NWORK | (1) | | | |
| CO31 | 2 | 0 | RATINT | (1) | 1 | ISYNC | (1) | | | |
| CO36 | 7 | 0 | IOCU | (5) | 5 | NONDED | (1) | 6 | NWOIO | (1) |
| CO42 | 3 | 0 | MISTKI | (3) | | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 216B | 142 |
| COMMON LENGTH | 2413B | 1291 |

```
      SUBROUTINE STATE2(NEXT)
C                        THIS VERSION:   MARCH 1976
C THIS SUBROUTINES SIMULATES THE RECOVERY PROCEDURE (ROLLAHEAD)
C
C********** ********* ********* ********* ********* ********* ********
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/COM7/REASON
      COMMON/COM8/FOCO
      COMMON/CO11/MISTAK
      COMMON/CO15/NOON(5),NWORK
      COMMON/CO17/NTRI,NQUA
      COMMON/CO18/IREP
      COMMON/CO28/MODSIM,NSPB
      COMMON/CO31/RATINT,ISYNC
      COMMON/CO41/ICATAS,I3
      COMMON/CO42/MISTKI(3)
      COMMON/PERM/LAST(5),MININT,PSUC
      COMMON/PHILM3/NTR3,NTR2,NTR1,NTNR1
      COMMON/RAHEAD/RACPU
      INTEGER EXTEN,PTR
      INTEGER REASON
      REAL LAST,MININT
C********** ********* ********* ********* ********* ********* ********
C
C IREP:   COMPUTER IN CHARGE OF THE REPAIR
      IREP=IRAN(1,NWORK-1)
   40 CONTINUE
      IF ((NOON(IREP).EQ.1).AND.(IREP.NE.INT(FOCO))) GO TO 30
      IREP=IREP+1
      IF (IREP.GT.MODSIM) IREP=1
      GO TO 40
   30 CONTINUE
      IDELET=0
C IF THE FAULT IS NOT RECURRENT, GO TO 10
      IF (TIME-LAST(FOCO).GT.MININT) GO TO 10
      NEXT=7
      RETURN
   10 CONTINUE
C IF THERE IS NO OTHER FAULT, GO TO 20
      IF (TABLE(PTR,1).GT.TIME+RECOV) GO TO 20
      CALL FIFAU(IN,NEXT,1)
      IF(NEXT.EQ.5) RETURN
      GOTO 10
C
   20 CONTINUE
C AFAULT IN AN OTHER COMPUTER PREEMPTS THE FIRST ONE WHICH IS SEEN AS A
C PERMANENT
      IDE=0
      DET=TIME+RECOV
      DO 21 I=1,IDIM
      IF (ACFAU(I,3).EQ.0.) GO TO 22
      IF (ACFAU(I,4).EQ.FOCO) GO TO 21
      IF (ACFAU(I,6).GE.DET) GO TO 21
      DET=ACFAU(I,6)
```

```
                IDE=I
              21 CONTINUE
              22 CONTINUE
            C IF NO OTHER FAULT, GO TO 23. ELSE DISCONNECT ONE COMPUTER.
60              IF (IDE.EQ.0) GO TO 23
                NOON(FOCO)=0
                NHORK=NHORK-1
                TIME=DET
                NEXT=1
65              IF (NHORK.EQ.4) NQUA=NQUA+1
                IF (NHORK.EQ.3) NTRI=NTRI+1
                IF (NHORK.EQ.2) NEXT=4
                RETURN
              23 CONTINUE
70              IGOOD=1
            C IS THE REPAIRING COMPUTER GOOD.
                DO 75 J=1,IDIM
                IF (ACFAU(J,3).EQ.0.) GO TO 76
                JJ=J
75              IF (INT(ACFAU(J,4)).EQ. IREP) IGOOD=0
              75 CONTINUE
              76 CONTINUE
                TIMS=TIME+RECOV
            C ERASING AND UPDATING OF DETECTION TIME
80              IEFFA=0
                DO 80 J=1,IDIM
                ACF3=ACFAU(J,3)
                IF (ACF3.EQ.0.) GO TO 81
            C  IF THE FAULT IS NOT DETECTED, GO  TO 90
85              IF ((ACFAU(J,6).GT.TIME-DELAY).OR.(ACFAU(J,4).NE.FOCO)) GO TO 90
            C IF THE FAULT LASTS TOO LONG, GO TO 101
                IF (ACFAU(J,2).GT.TIME) GO TO 101
                IF(RANF(0.0).GE.RACPU) GOTO 101
                IF (ACFAU(J,5).NE.0.) GO TO 101
90              IF (IGOOD.EQ.0) GO TO 50
                IDELET=1
                IF (ISYNC.EQ.1) GO TO 1111
            C ASYNCHRONOUS CASE. DO NOT COUNT A SUCCESSFUL RECOVERY. IF ALL PSEUDO-
            C FAULTS HAVE NOT YET BEEN ERASED.
95              DO  1110 I=1,JJ
                IF (ACFAU(I,3).EQ.0.) GO TO 1110
                IF (I.EQ.J) GO TO 1110
                IF (ACFAU(I,2).NE.ACFAU(J,2)) GO TO 1110
                IDELET=IDELET-1
100             NTR3=NTR3-1
                GO TO 1111
             1110 CONTINUE
             1111 CONTINUE
                IFFFA=1
105             ACFAU(J,3)=0.
                NTR3 = NTR3 + 1                                          OL29APR4
                GOTO 80
            C
              50 CONTINUE
110             ACFAU(J,6)=ACFAU(J,6)+TC
```

```
                GO TO 80
             90 CONTINUE
          C UPDATING FOR THE NON DETECTED FAULTS
                ACFAU(J,6)=AMAX1(TIMF-DELAY+TC*(1.+AINT((DELAY+RECOV) /TC))
115            1                 ,ACFAU(J,6))
                GO TO 80
            101 CONTINUE
                CALL DETTIM(DETEC,TIMS,ACFAU(J,3))
                ACFAU(J,6)=DETEC
120          80 CONTINUE
             81 CONTINUE
                IF (IEFFA.EQ.1) CALL GATHER
                NEXT=1
                TIME=TIMS
125             LAST(FOCO)=TIME
                CALL MISCYC(TIME-RECOV,TIME,NEXT)
                IF (NEXT.NE.5) RETURN
                ICATAS=ICATAS+1
                REASON=6
130             IF (IDELET.EQ.0) RETURN
                NTR3=NTR3-1
                MISTAK=MISTAK+1
                MISTKI(NWORK-2)=MISTKI(NWORK-2)+1
                END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | STATE2 | 1 | 38 | 43 | 68 | 127 | 130 | 134 | | |

| VARIABLES | | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | | REAL | ARRAY | COM3 | REFS | 7 | 52 | 53 | 54 | 55 | 73 | 75 |
| | | | | | | 82 | 2*85 | 87 | 89 | 96 | 2*98 | 110 | 114 |
| | | | | | | 118 | DEFINED | 105 | 110 | 114 | 119 | | |
| 340 | ACF3 | | REAL | | | REFS | 83 | DEFINED | 82 | | | | |
| 2265 | DELAY | | REAL | | COM1 | REFS | 6 | 85 | 2*114 | | | | |
| 331 | DET | | REAL | | | REFS | 54 | 63 | DEFINED | 50 | 55 | | |
| 341 | DETEC | | REAL | | | REFS | 118 | 119 | | | | | |
| 74 | ENDMIS | | REAL | | COM3 | REFS | 7 | | | | | | |
| 2262 | EXTEN | | INTEGER | | COM1 | REFS | 6 | 21 | | | | | |
| 0 | FOCO | | REAL | | COM8 | REFS | 9 | 29 | 36 | 53 | 61 | 85 | 125 |
| 332 | I | | INTEGER | | | REFS | 52 | 53 | 54 | 55 | 56 | 96 | 97 |
| | | | | | | 98 | DEFINED | 51 | 95 | | | | |
| 0 | ICATAS | | INTEGER | | C041 | REFS | 16 | 128 | DEFINED | 128 | | | |
| 330 | IDE | | INTEGER | | | REFS | 60 | DEFINED | 49 | 56 | | | |
| 326 | IDELET | | INTEGER | | | REFS | 99 | 130 | DEFINED | 34 | 91 | 99 | |
| 2263 | IDETEC | | INTEGER | | COM1 | REFS | 6 | | | | | | |
| 0 | IDIM | | INTEGER | | COM1 | REFS | 6 | 51 | 72 | 81 | | | |
| 337 | IEFFA | | INTEGER | | | REFS | 122 | DEFINED | 80 | 104 | | | |
| 333 | IGOOD | | INTEGER | | | REFS | 90 | DEFINED | 70 | 75 | | | |
| 327 | IN | * | INTEGER | | | REFS | 42 | | | | | | |
| 0 | IREP | | INTEGER | | C018 | REFS | 13 | 2*29 | 30 | 31 | 75 | | |
| | | | | | | DEFINED | 27 | 30 | 31 | | | | |
| 1 | ISYNC | | INTEGER | | C031 | REFS | 15 | 92 | | | | | |
| 1 | I3 | | INTEGER | | C041 | REFS | 16 | | | | | | |
| 334 | J | | INTEGER | | | REFS | 73 | 74 | 75 | 82 | 2*85 | 87 | 89 |
| | | | | | | 97 | 98 | 105 | 2*110 | 2*114 | 118 | 119 | |
| | | | | | | DEFINED | 72 | 81 | | | | | |
| 335 | JJ | | INTEGER | | | REFS | 95 | DEFINED | 74 | | | | |
| 0 | LAST | | REAL | ARRAY | PERM | REFS | 18 | 23 | 36 | DEFINED | 125 | | |
| 75 | MEMSIZ | | INTEGER | | COM3 | REFS | 7 | | | | | | |
| 5 | MININT | | REAL | | PERM | REFS | 18 | 23 | 36 | | | | |
| 0 | MISTAK | | INTEGER | | C011 | REFS | 10 | 132 | DEFINED | 132 | | | |
| 0 | MISTKI | | INTEGER | ARRAY | C042 | REFS | 17 | 133 | DEFINED | 133 | | | |
| 0 | MODSIM | | INTEGER | | C028 | REFS | 14 | 31 | | | | | |
| 0 | NEXT | | INTEGER | | F.P. | REFS | 42 | 43 | 126 | 127 | DEFINED | 1 | 37 |
| | | | | | | 64 | 67 | 123 | | | | | |
| 0 | NOON | | INTEGER | ARRAY | C015 | REFS | 11 | 29 | DEFINED | 61 | | | |
| 1 | NQUA | | INTEGER | | C017 | REFS | 12 | 65 | DEFINED | 65 | | | |
| 1 | NSPB | | INTEGER | | C028 | REFS | 14 | | | | | | |
| 3 | NTNR1 | | INTEGER | | PHILM3 | REFS | 19 | | | | | | |
| 0 | NTRI | | INTEGER | | C017 | REFS | 12 | 66 | DEFINED | 66 | | | |
| 2 | NTR1 | | INTEGER | | PHILM3 | REFS | 19 | | | | | | |
| 1 | NTR2 | | INTEGER | | PHILM3 | REFS | 19 | | | | | | |
| 0 | NTR3 | | INTEGER | | PHILM3 | REFS | 19 | 100 | 106 | 131 | DEFINED | 100 | 106 |
| | | | | | | 131 | | | | | | | |
| 5 | NWORK | | INTEGER | | C015 | REFS | 11 | 27 | 62 | 65 | 66 | 67 | 2*133 |
| | | | | | | DEFINED | 62 | | | | | | |
| 6 | PSUC | | REAL | | PERM | REFS | 18 | | | | | | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2261 | PTR | INTEGER | | COM1 | REFS | 6 | 21 | 41 | | | |
| 0 | RACPU | REAL | | RAHEAD | REFS | 20 | 88 | | | | |
| 0 | RATINT | REAL | | CO31 | REFS | 15 | | | | | |
| 0 | REASON | INTEGER | | COM7 | REFS | 8 | 22 | DEFINED | 129 | | |
| 2264 | RECOV | REAL | | COM1 | REFS | 6 | 41 | 50 | 78 | 114 | 126 |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 6 | 41 | | | | |
| 76 | TC | REAL | | COM3 | REFS | 7 | 110 | 2*114 | | | |
| 2266 | TIME | REAL | | COM1 | REFS | 6 | 36 | 41 | 50 | 78 | 85 87 |
| | | | | | | 114 | 125 | 2*126 | DEFINED | 63 | 124 |
| 336 | TIMS | REAL | | | REFS | 118 | 124 | DEFINED | 78 | | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| DETTIM | | 3 | 118 |
| FIFAU | | 3 | 42 |
| GATHER | | 0 | 122 |
| IRAN | INTEGER | 2 | 27 |
| MISCYC | | 3 | 126 |
| RANF | REAL | 1 | 88 |

| INLINE FUNCTIONS | TYPE | ARGS | | DEF LINE | REFERENCES | | | |
|---|---|---|---|---|---|---|---|---|
| AINT | REAL | 1 | INTRIN | | 114 | | | |
| AMAX1 | REAL | 0 | INTRIN | | 114 | | | |
| INT | INTEGER | 1 | INTRIN | | 29 | 75 | | |

| STATEMENT LABELS | | DEF LINE | REFERENCES | | | | |
|---|---|---|---|---|---|---|---|
| 33 | 10 | 39 | 36 | 44 | | | |
| 47 | 20 | 46 | 41 | | | | |
| 67 | 21 | 57 | 51 | 53 | 54 | | |
| 71 | 22 | 58 | 52 | | | | |
| 117 | 23 | 69 | 60 | | | | |
| 23 | 30 | 33 | 29 | | | | |
| 10 | 40 | 28 | 32 | | | | |
| 217 | 50 | 109 | 90 | | | | |
| 0 | 75 | 76 | 72 | | | | |
| 135 | 76 | 77 | 73 | | | | |
| 241 | 80 | 120 | 81 | 107 | 111 | 116 | |
| 244 | 81 | 121 | 83 | | | | |
| 222 | 90 | 112 | 85 | | | | |
| 234 | 101 | 117 | 87 | 88 | 89 | | |
| 211 | 1110 | 102 | 95 | 96 | 97 | 98 | |
| 213 | 1111 | 103 | 92 | 101 | | | |

| LOOPS | LABEL | | INDEX | FROM-TO | LENGTH | PROPERTIES | | | |
|---|---|---|---|---|---|---|---|---|---|
| 57 | 21 | * | I | 51 57 | 120 | OPT | EXITS | | |
| 124 | 75 | * | J | 72 76 | 118 | OPT | EXITS | | |
| 141 | 80 | * | J | 81 120 | 1030 | EXT REFS | EXITS | NOT INNER | |
| 177 | 1110 | * | I | 95 102 | 140 | OPT | EXITS | | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 IDIM (1) | 1 TABLE (1200) | 1201 PTR (1) | | | |
| | | 1202 EXTEN (1) | 1203 IDETEC (1) | 1204 RECOV (1) | | | |
| | | 1205 DELAY (1) | 1206 TIME (1) | | | | |
| COM3 | 63 | 0 ACFAU (60) | 60 ENDMIS (1) | 61 MEMSIZ (1) | | | |
| | | 62 TC (1) | | | | | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| COM7 | 1 | 0 | REASON | (1) | | | | | | |
| COM8 | 1 | 0 | FOCO | (1) | | | | | | |
| CO11 | 1 | 0 | HISTAK | (1) | | | | | | |
| CO15 | 6 | 0 | NOON | (5) | 5 | NWORK | (1) | | | |
| CO17 | 2 | 0 | NTRI | (1) | 1 | NQUA | (1) | | | |
| CO18 | 1 | 0 | IREP | (1) | | | | | | |
| CO28 | 2 | 0 | MOOSIM | (1) | 1 | NSPB | (1) | | | |
| CO31 | 2 | 0 | RATINT | (1) | 1 | ISYNC | (1) | | | |
| CO41 | 2 | 0 | ICATAS | (1) | 1 | I3 | (1) | | | |
| CO42 | 3 | 0 | HISTKI | (3) | | | | | | |
| PERM | 7 | 0 | LAST | (5) | 5 | MININT | (1) | 6 | PSUC | (1) |
| PHILM3 | 4 | 0 | NTR3 | (1) | 1 | NTR2 | (1) | 2 | NTR1 | (1) |
| | | 3 | NTNR1 | (1) | | | | | | |
| RAHEAD | 1 | 0 | RACPU | (1) | | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 342B | 226 |
| COMMON LENGTH | 24278 | 1303 |

```
      SUBROUTINE STATE7(NEXT)
C                    THIS VERSION:  MARCH 1976
C SIMULATION OF A SYSTEM RESTART
C
C ******************************************************************
      COMMON/COM1/IOIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM7/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/COM7/REASON
      COMMON/COM9/COPLAS(5),RMC,DURMC
      COMMON/CO11/MISTAK
      COMMON/CO13/DURRES
      COMMON/CO15/NOON(5),NWORK
      COMMON/CO41/ICATAS,I3
      COMMON/CO42/MISTKI(3)
      COMMON/PERM/LAST(5),MININT,PSUC
      COMMON/PHILM3/NTR3,NTR2,NTR1,NTNR1
      INTEGER PTR,EXTEN,REASON
      REAL LAST,MININT
C********** ********* ********* ********** ********* ********* ********
C
   10 CONTINUE
      IF (TABLE(PTR,1).GT.TIME+DURRES) GO TO 20
      CALL FIFAU(IN,NEXT)
      IF(NEXT.EQ.5) RETURN
      GOTO 10
   20 CONTINUE
      IRA=0
      DO 60 J=1,IOIM
      IF (ACFAU(J,3).EQ.0.) GO TO 61
      IF((ACFAU(J,2).LT.TIME).OR. ((ACFAU(J,2).LT.TIME+DURRES).AND.
     1                              (ACFAU(J,5).EQ.0.))) ACFAU(J,3)=0.
      IF (ACFAU(J,3).NE.0.) GO TO 60
      IRA=IRA+1
      NTR3=NTR3+1
   60 CONTINUE
   61 CONTINUE
      IF (IRA.GE.1) CALL GATHER
      TIME=TIME+DURRES
      DO 100 J=1,IOIM
      ACF3=ACFAU(J,3)
      IF (ACF3.EQ.0.) GO TO 110
      CALL DETTIM(ACFAU(J,6),TIME,ACF3)
  100 CONTINUE
  110 NEXT=1
      DO 120 I=1,5
      COPLAS(I)=TIME
      LAST(I)=TIME
  120 CONTINUE
      CALL MISCYC(TIME-DURRES,TIME,NEXT)
      IF (NEXT.NE.5) RETURN
      ICATAS=ICATAS+1
      REASON=6
      NTR3=NTR3-IRA
      MISTAK=MISTAK+IRA
      MISTKI(NWORK-2)=MISTKI(NWORK-2)+IRA
```

```
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | | |
|---|---|---|---|---|---|---|
| 2 | STATE3 | 1 | 24 | 50 | 56 | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 7 | 29 | 3*30 | 32 | 40 | 42 |
| | | | | | DEFINED | 30 | | | | | |
| 136 | ACF3 | REAL | | | REFS | 41 | 42 | DEFINED | 40 | | |
| 0 | COPLAS | REAL | ARRAY | COM9 | REFS | 9 | DEFINED | 46 | | | |
| 2265 | DELAY | REAL | | COM1 | REFS | 6 | | | | | |
| 6 | DURMC | REAL | | COM9 | REFS | 9 | | | | | |
| 0 | DURRES | REAL | | C013 | REFS | 11 | 22 | 30 | 38 | 49 | |
| 74 | ENDMIS | REAL | | COM3 | REFS | 7 | | | | | |
| 2262 | EXTEN | INTEGER | | COM1 | REFS | 6 | 17 | | | | |
| 137 | I | INTEGER | | | REFS | 46 | 47 | DEFINED | 45 | | |
| 0 | ICATAS | INTEGER | | C041 | REFS | 13 | 51 | DEFINED | 51 | | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 6 | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 6 | 28 | 39 | | | |
| 133 | IN | * INTEGER | | | REFS | 23 | | | | | |
| 134 | IRA | INTEGER | | | REFS | 33 | 37 | 53 | 54 | 55 | |
| | | | | | DEFINED | 27 | 33 | | | | |
| 1 | I3 | INTEGER | | C041 | REFS | 13 | | | | | |
| 135 | J | INTEGER | | | REFS | 29 | 4*30 | 32 | 40 | 42 | |
| | | | | | DEFINED | 28 | 39 | | | | |
| 0 | LAST | REAL | ARRAY | PERM | REFS | 15 | 18 | DEFINED | 47 | | |
| 75 | MEMSIZ | INTEGER | | COM3 | REFS | 7 | | | | | |
| 5 | MININT | REAL | | PERM | REFS | 15 | 18 | | | | |
| 0 | MISTAK | INTEGER | | C011 | REFS | 10 | 54 | DEFINED | 54 | | |
| 0 | MISTKI | INTEGER | ARRAY | C042 | REFS | 14 | 55 | DEFINED | 55 | | |
| 0 | NEXT | INTEGER | | F.P. | REFS | 23 | 24 | 49 | 50 | DEFINED | 1 44 |
| 0 | NOON | INTEGER | ARRAY | C015 | REFS | 12 | | | | | |
| 3 | NTNR1 | INTEGER | | PHILM3 | REFS | 16 | | | | | |
| 2 | NTR1 | INTEGER | | PHILM3 | REFS | 16 | | | | | |
| 1 | NTR2 | INTEGER | | PHILM3 | REFS | 16 | | | | | |
| 0 | NTR3 | INTEGER | | PHILM3 | REFS | 16 | 34 | 53 | DEFINED | 34 | 53 |
| 5 | NWORK | INTEGER | | C015 | REFS | 12 | 2*55 | | | | |
| 6 | PSUC | REAL | | PERM | REFS | 15 | | | | | |
| 2261 | PTR | INTEGER | | COM1 | REFS | 6 | 17 | 22 | | | |
| 0 | REASON | INTEGER | | COM7 | REFS | 8 | 17 | DEFINED | 52 | | |
| 2264 | RECOV | REAL | | COM1 | REFS | 6 | | | | | |
| 5 | RMC | REAL | | COM9 | REFS | 9 | | | | | |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 6 | 22 | | | | |
| 76 | TC | REAL | | COM3 | REFS | 7 | | | | | |
| 2266 | TIME | REAL | | COM1 | REFS | 6 | 22 | 2*30 | 38 | 42 | 46 47 |
| | | | | | 2*49 | DEFINED | 38 | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| GETTIM | | 3 | 42 |
| FIFAU | | 2 | 23 |
| GATHER | | 0 | 37 |
| MISCYC | | 3 | 49 |

| STATEMENT LABELS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|
| 4 | 10 | 21 | 25 | |
| 20 | 20 | 26 | 22 | |
| 42 | 60 | 35 | 28 | 32 |
| 44 | 61 | 36 | 29 | |
| 0 | 100 | 43 | 39 | |
| 65 | 110 | 44 | 41 | |
| 0 | 120 | 48 | 45 | |

| LOOPS | LABEL | | INDEX | FROM-TO | LENGTH | PROPERTIES | EXITS | |
|---|---|---|---|---|---|---|---|---|
| 26 | 60 | * | J | 28 35 | 168 | OPT | EXT REFS | EXITS |
| 53 | 100 | * | J | 39 43 | 128 | | | |
| 71 | 120 | | I | 45 48 | 28 | INSTACK | | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 | IDIM | (1) | 1 | TABLE | (1200) | 1201 | PTR | (1) |
| | | 1202 | EXTEN | (1) | 1203 | IDETEC | (1) | 1204 | RECOV | (1) |
| | | 1205 | DELAY | (1) | 1206 | TIME | (1) | | | |
| COM3 | 63 | 0 | ACFAU | (60) | 60 | ENDMIS | (1) | 61 | MEMSIZ | (1) |
| | | 62 | TC | (1) | | | | | | |
| COM7 | 1 | 0 | REASON | (1) | | | | | | |
| COM9 | 7 | 0 | COPLAS | (5) | 5 | RMC | (1) | 6 | DURMC | (1) |
| CO11 | 1 | 0 | MISTAK | (1) | | | | | | |
| CO13 | 1 | 0 | DURRES | (1) | | | | | | |
| CO15 | 6 | 0 | NOON | (5) | 5 | NWORK | (1) | | | |
| CO41 | 2 | 0 | ICATAS | (1) | 1 | I3 | (1) | | | |
| CO42 | 3 | 0 | MISTKI | (3) | | | | | | |
| PERM | 7 | 0 | LAST | (5) | 5 | MININT | (1) | 6 | PSUC | (1) |
| PHILM3 | 4 | 0 | NTR3 | (1) | 1 | NTR2 | (1) | 2 | NTR1 | (1) |
| | | 3 | NTNR1 | (1) | | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 1408 | 96 |
| COMMON LENGTH | 24268 | 1302 |

```
      SUBROUTINE STATE4(NEXT)
C                        THIS VERSION:   MARCH 1976
C SIMULATION OF DUPLEX
C
C ******************************************************************
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/COM7/REASON
      COMMON/CO11/MISTAK
      COMMON/CO15/NOON(5),NWORK
      COMMON/CO31/RATINT,ISYNC
      COMMON/CO38/IOCU(5),NONDED,NWOIO
      COMMON/CO42/MISTKI(3)
      INTEGER PTR,EXTEN
      INTEGER REASON
C********** ********* ********** ********** ********* ********* ********
C
C SUPPRESSION OF THE FAULTS IN THE SWITCHED-OFF COMPUTER
      IRA=0
      DO 10 J=1,IDIM
      IF (ACFAU(J,3).EQ.0.) GO TO 20
      IF (NOON(ACFAU(J,4)).EQ.1) GO TO 10
C TEST IF A TRANSIENT IS MISTAKEN AS A PERMANENT
      ACFAU(J,3)=0.
      IRA=IRA+1
      IF((ACFAU(J,2).GE.ENDMIS).OR.(IRA.NE.1)) GOTO 10
      MISTAK=MISTAK+1
      MISTKI(1)=MISTKI(1)+1
   10 CONTINUE
   20 CONTINUE
      IF (IRA.GE.1) CALL GATHER
C IF THERE  IS A LURKING FAULT GO TO 30.
      IF (ACFAU(1,3).NE.0.) GO TO 30
C IF THERE IS NO MORE FAULT, RETURN
   70 CONTINUE
      TIME=TABLE(PTR,1)
      IF (TIME.LT.ENDMIS) GO TO 40
   50 CONTINUE
      NEXT=6
      RETURN
   40 CONTINUE
      CALL FIFAU(IN,NEXT,ISYNC)
      IF(NEXT.EQ.5) RETURN
      IF(IN.EQ.0) GOTO 70
C DETERMINE DETECTION TIME
   30 CONTINUE
C        THE FAMOUS ONE TRILLION APPROACH AGAIN, SPORTSFANS ...          SP29APR4
      DET=1000000000000.                                                 SP29APR4
      DO 90 I=1,IDIM
      IF (ACFAU(I,3).EQ.0.) GO TO 100
      IF (ACFAU(I,6).GE.DET ) GO TO 90
      DET=ACFAU(I,6)
      IDETEC=I
   90 CONTINUE
  100 CONTINUE
```

```
      C IF THERE IS NO FAULT BEFORE DFT, START ROLLBACK.
            TIME=AMIN1(TABLE(PTR,1),DFT)
            IF (TIME.GE.ENDMIS) GO TO 50
            IF (TIME.NE.DET) GO TO 110
60          NEXT=8
            RETURN
          110 CONTINUE
            CALL FIFAU(IN,NEXT,ISYNC)
            IF(NEXT.EQ.5) RETURN
65          IF(IN.EQ.0) GOTO 100
            GOTO 30
            END
```

SYMBOLIC REFERENCE MAP .

| ENTRY POINTS | | DEF LINE | REFERENCES | | | |
|---|---|---|---|---|---|---|
| 2 | STATE4 | 1 | 40 | 43 | 61 | 64 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 7 | 21 | 22 | 26 | 33 | 50 | 51 |
| | | | | | | 52 | DEFINED | 24 | | | | |
| 2265 | DELAY | REAL | | COM1 | REFS | 6 | | | | | | |
| 140 | DET | REAL | | | REFS | 51 | 57 | 59 | DEFINED | 48 | 52 | |
| 74 | ENDMIS | REAL | | COM3 | REFS | 7 | 26 | 37 | 58 | | | |
| 2262 | EXTEN | INTEGER | | COM1 | REFS | 6 | 14 | | | | | |
| 141 | I | INTEGER | | | REFS | 50 | 51 | 52 | 53 | DEFINED | 49 | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 6 | DEFINED | 53 | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 6 | 20 | 49 | | | | |
| 137 | IN | INTEGER | | | REFS | 42 | 44 | 63 | 65 | | | |
| 0 | IOCU | INTEGER | ARRAY | C038 | REFS | 12 | | | | | | |
| 135 | IRA | INTEGER | | | REFS | 25 | 26 | 31 | DEFINED | 19 | 25 | |
| 1 | ISYNC | INTEGER | | C031 | REFS | 11 | 42 | 63 | | | | |
| 136 | J | INTEGER | | | REFS | 21 | 22 | 24 | 26 | DEFINED | 20 | |
| 75 | MEMSIZ | INTEGER | | COM3 | REFS | 7 | | | | | | |
| 0 | MISTAK | INTEGER | | C011 | REFS | 9 | 27 | DEFINED | 27 | | | |
| 0 | MISTKI | INTEGER | ARRAY | C042 | REFS | 13 | 28 | DEFINED | 28 | | | |
| 0 | NEXT | INTEGER | | F.P. | REFS | 42 | 43 | 63 | 64 | DEFINED | 1 | 39 |
| | | | | | | 60 | | | | | | |
| 5 | NONDED | INTEGER | | C038 | REFS | 12 | | | | | | |
| 0 | NOON | INTEGER | ARRAY | C015 | REFS | 10 | 22 | | | | | |
| 6 | NHOIO | INTEGER | | C038 | REFS | 12 | | | | | | |
| 5 | NWORK | INTEGER | | C015 | REFS | 10 | | | | | | |
| 2261 | PTR | INTEGER | | COM1 | REFS | 6 | 14 | 36 | 57 | | | |
| 0 | RATINT | REAL | | C031 | REFS | 11 | | | | | | |
| 0 | REASON | INTEGER | | COM7 | REFS | 8 | 15 | | | | | |
| 2264 | RECOV | REAL | | COM1 | REFS | 6 | | | | | | |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 6 | 36 | 57 | | | | |
| 76 | TC | REAL | | COM3 | REFS | 7 | | | | | | |
| 2266 | TIME | REAL | | COM1 | REFS | 6 | 37 | 58 | 59 | DEFINED | 36 | 57 |

| EXTERNALS | TYPE | ARGS | REFERENCES | | |
|---|---|---|---|---|---|
| FIFAU | | 3 | 42 | 63 | |
| GATHER | | 0 | 31 | | |

| INLINE FUNCTIONS | TYPE | ARGS | DEF LINE | REFERENCES |
|---|---|---|---|---|
| AMIN1 | REAL | 0 | INTRIN | 57 |

| STATEMENT LABELS | | DEF LINE | REFERENCES | | |
|---|---|---|---|---|---|
| 27 | 10 | 29 | 20 | 22 | 26 |
| 31 | 20 | 30 | 21 | | |
| 62 | 30 | 46 | 33 | 66 | |
| 47 | 40 | 41 | 37 | | |
| 44 | 50 | 38 | 58 | | |
| 40 | 70 | 35 | 44 | | |
| 77 | 90 | 54 | 49 | 51 | |
| 101 | 100 | 55 | 50 | 65 | |
| 114 | 110 | 62 | 59 | | |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | EXITS |
|---|---|---|---|---|---|---|
| 12 | 10 | * J | 20 29 | 178 | OPT | EXITS |
| 71 | 90 | * I | 49 54 | 108 | OPT | EXITS |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS | NAME | (LENGTH) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 | IDTM | (1) | 1 | TABLE | (1200) | 1201 | PTR | (1) |
| | | 1202 | EXTEN | (1) | 1203 | IDETEC | (1) | 1204 | RECOV | (1) |
| | | 1205 | DELAY | (1) | 1206 | TIME | (1) | | | |
| COM3 | 63 | 0 | ACFAU | (60) | 60 | ENDMIS | (1) | 61 | NEMSIZ | (1) |
| | | 62 | TC | (1) | | | | | | |
| COM7 | 1 | 0 | REASON | (1) | | | | | | |
| C011 | 1 | 0 | MISTAK | (1) | | | | | | |
| C015 | 6 | 0 | NOON | (5) | 5 | NWORK | (1) | | | |
| C031 | 2 | 0 | RATINT | (1) | 1 | ISYNC | (1) | | | |
| C038 | 7 | 0 | IOCU | (5) | 5 | NONDED | (1) | 6 | NWOIO | (1) |
| C042 | 3 | 0 | MISTKT | (3) | | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 1428 | 98 |
| COMMON LENGTH | 2412B | 1290 |

```
      SUBROUTINE STATE7(NEXT)
C                        THIS VERSION:   MARCH 1976
C THIS IS A MEMORY COPY. THE COMPUTER IN CHARGE OF THE REPAIR IS IREP.
C IF A NEW FAULT HIT THE SYSTEM BEFORE THE END OF THE MEMORY COPY, THE
C FIRST FAULT IS CONSIDERED AS A PERMANENT.
C
C ***************************************************************************
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/COM7/REASON
      COMMON/COM8/FOCO
      COMMON/COM9/COPLAS(5),RMC,DURMC
      COMMON/CO11/MISTAK
      COMMON/CO15/NOON(5),NWORK
      COMMON/CO17/NTRI,NQUA
      COMMON/CO18/IREP
      COMMON/CO28/MODSIM,NSPB
      COMMON/CO29/NMC
      COMMON/CO31/RATINT,ISYNC
      COMMON/CO41/ICATAS,I3
      COMMON/CO42/MISTKI(3)
      COMMON/CO45/PSMC
      COMMON/PERM/LAST(5),MININT,PSUC
      COMMON/PHILM3/NTR3,NTR2,NTR1,NTNR1
      INTEGER EXTEN,PTR,REASON
      REAL LAST,MININT
C********** ********* ********* ********** ********* ********* ********
C
      IDELET=0
C IF THE FAULT IS NOT RECURRENT, GO TO 10.
      IF (TIME-COPLAS(FOCO).GT.RMC) GO TO 10
      NMC=NMC-1
      NOON(FOCO)=0
      IF (NSPB.NE.0) GO TO 24
      NWORK=NWORK-1
      NEXT=1
      IF (NWORK.EQ.4)NQUA=NQUA+1
      IF (NWORK.EQ.3)NTRI=NTRI+1
      IF (NWORK.LT.3) NEXT=4
      RETURN
   24 CONTINUE
      NEXT=12
      RETURN
   10 CONTINUE
C IF THERE IS NO OTHER FAULT, GO TO 20
      IF (TABLE(PTR,1).GT.TIME+DURMC) GO TO 20
      CALL FIFAU(IN,NEXT,1)
      IF(NEXT.EQ.5) RETURN
      GOTO 10
   20 CONTINUE
C A FAULT IN ANOTHE R COMPUTER PREEMPTS THE FIRST ONE WHICH IS SEEN AS A
C PERMANENT
      IDE=0
      DFT=TIME+DURMC
      DO 21 I=1,IDIM
```

```
                IF (ACFAU(I,3).EQ.0.) GO TO 22
                IF (ACFAU(I,4).EQ.FOCO) GO TO 21
                IF (ACFAU(I,6).GE.DET) GO TO 21
                DET=ACFAU(I,6)
60              IDE=I
             21 CONTINUE
             22 CONTINUE
                IF (IDE.EQ.0) GO TO 23
                NOON(FOCO)=0
65              NWORK=NWORK-1
                TIME=DET
                NEXT=1
                IF (NWORK.EQ.4) NQUA=NQUA+1
                IF (NWORK.EQ.3) NTRI=NTRI+1
70              IF (NWORK.EQ.2) NEXT=4
                RETURN
             23 CONTINUE
                MUL=0
                DO 40 I=1,IDIM
75              IF (ACFAU(I,3).EQ.0.) GO TO 50
                JJ=I
                IF (INT(ACFAU(I,4)).EQ.IREP) MUL=I
             40 CONTINUE
             50 CONTINUE
80           C ERASING OF THE FAULT
                IEFFA=0
                DO 60 J=1,IDIM
                IF (ACFAU(J,3).EQ.0.) GO TO 61
             C ERASED FAULTS : MEMORY-FAULT DYING BEFORE BEGINNING OF THE CORRECTION
85           C                 OTHER FAULT (EXTENTO) DYING BEFORE END OF CORRECTION
                IF ((ACFAU(J,2).LT.TIME).OR.((ACFAU(J,2).LT.TIME+DURMC).AND.
               1                             (ACFAU(J,5).EQ.0.))) GO TO 70        0L29APR4
                GO TO 60
             90 CONTINUE
90              ACFAU(J,5)=ACFAU(J,5)+ACFAU(MUL,5)
                IF (ACFAU(MUL,3).EQ.3.) ACFAU(J,3)=3.
                GO TO 60
             70 CONTINUE                                                         0L29APR4
                IF (RANF(0.).GT.PSMC) GO TO 61
95              IF (MUL.NE.0) GO TO 90
                IDELET=1
                IF (ISYNC.EQ.1) GO TO 1111
                DO  1110 I=1,JJ
                IF (ACFAU(I,3).EQ.0.) GO TO 1110
100             IF (I.EQ.J) GO TO 1110
                IF (ACFAU(I,2).NE.ACFAU(J,2)) GO TO 1110
                IDELET=IDELET-1
                NTR3=NTR3-1
                GO TO 1111
105          1110 CONTINUE
             1111 CONTINUE
                IEFFA=1
                ACFAU(J,3) = 0.                                                  0L29APR4
                NTR3 = NTR3 + 1                                                  0L29APR4
110          60 CONTINUE
```

```
   61 CONTINUE
      IF (IFFFA.EQ.1) CALL GATHER
C UPDATING DETECTION TIME
      TIME=TIME+DURMC
115   DO 100 J=1,IDIM
      IF (ACFAU(J,3).EQ.0.) GO TO 110
      CALL DETTIM(DETEC,TIME,ACFAU(J,3))
      ACFAU(J,6)=DETEC
  100 CONTINUE
120   110 NEXT=1
      COPLAS(FOCO)=TIME
      LAST(FOCO)=TIME
      CALL MISCYC(TIME-RECOV,TIME,NEXT)
      IF (NEXT.NE.5) RETURN
125   ICATAS=ICATAS+1
      REASON=6
      IF (IDELET.EQ.0) RETURN
      NTR3=NTR3-1
      MISTAK=MISTAK+1
130   MISTKI(NWORK-2)=MISTKI(NWORK-2)+1
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | STATE7 | 1 | 40 | 43 | 48 | 71 | 124 | 127 | 131 | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 9 | 56 | 57 | 58 | 59 | 75 | 77 |
| | | | | | 83 | 3*86 | 2*90 | 91 | 99 | 2*101 | 116 | 117 |
| | | | | | DEFINED | 90 | 91 | 108 | 118 | | | |
| 0 | COPLAS | REAL | ARRAY | COM9 | REFS | 12 | 31 | DEFINED | 121 | | | |
| 2265 | DELAY | REAL | | COM1 | REFS | 8 | | | | | | |
| 334 | DET | REAL | | | REFS | 58 | 66 | DEFINED | 54 | 59 | | |
| 342 | DETEC | REAL | | | REFS | 117 | 118 | | | | | |
| 6 | DURMC | REAL | | COM9 | REFS | 12 | 46 | 54 | 86 | 114 | | |
| 74 | ENDMIS | REAL | | COM3 | REFS | 9 | | | | | | |
| 2262 | EXTEN | INTEGER | | COM1 | REFS | 8 | 25 | | | | | |
| 0 | FOCO | REAL | | COM8 | REFS | 11 | 31 | 33 | 57 | 64 | 121 | 122 |
| 335 | I | INTEGER | | | REFS | 56 | 57 | 58 | 59 | 60 | 75 | 76 |
| | | | | | 2*77 | 99 | 100 | 101 | DEFINED | 55 | 74 | 98 |
| 0 | ICATAS | INTEGER | | C041 | REFS | 20 | 125 | DEFINED | 125 | | | |
| 333 | IDE | INTEGER | | | REFS | 63 | DEFINED | 53 | 60 | | | |
| 331 | IDELET | INTEGER | | | REFS | 102 | 127 | DEFINED | 29 | 96 | 102 | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 8 | | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 8 | 55 | 74 | 82 | 115 | | |
| 340 | IEFFA | INTEGER | | | REFS | 112 | DEFINED | 81 | 107 | | | |
| 332 | IN * | INTEGER | | | REFS | 47 | | | | | | |
| 0 | IREP | INTEGER | | C018 | REFS | 16 | 77 | | | | | |
| 1 | ISYNC | INTEGER | | C031 | REFS | 19 | 97 | | | | | |
| 1 | I3 | INTEGER | | C041 | REFS | 28 | | | | | | |
| 341 | J | INTEGER | | | REFS | 83 | 3*86 | 2*90 | 91 | 100 | 101 | 108 |
| | | | | | 116 | 117 | 118 | DEFINED | 82 | 115 | | |
| 337 | JJ | INTEGER | | | REFS | 98 | DEFINED | 76 | | | | |
| 0 | LAST | REAL | ARRAY | PERM | REFS | 23 | 26 | DEFINED | 122 | | | |
| 75 | MEMSIZ | INTEGER | | COM3 | REFS | 9 | | | | | | |
| 5 | MININT | REAL | | PERM | REFS | 23 | 26 | | | | | |
| 0 | MISTAK | INTEGER | | C011 | REFS | 13 | 129 | DEFINED | 129 | | | |
| 0 | MISTKI | INTEGER | ARRAY | C042 | REFS | 21 | 130 | DEFINED | 130 | | | |
| 0 | MODSIM | INTEGER | | C028 | REFS | 17 | | | | | | |
| 336 | MUL | INTEGER | | | REFS | 90 | 91 | 95 | DEFINED | 73 | 77 | |
| 0 | NEXT | INTEGER | | F.P. | REFS | 47 | 48 | 123 | 124 | DEFINED | 1 | 36 |
| | | | | | 39 | 42 | 67 | 70 | 120 | | | |
| 0 | NMC | INTEGER | | C029 | REFS | 18 | 32 | DEFINED | 32 | | | |
| 0 | NOON | INTEGER | ARRAY | C015 | REFS | 14 | DEFINED | 33 | 64 | | | |
| 1 | NOUA | INTEGER | | C017 | REFS | 15 | 37 | 68 | DEFINED | 37 | 68 | |
| 1 | NSPB | INTEGER | | C028 | REFS | 17 | 34 | | | | | |
| 3 | NTNR1 | INTEGER | | PHILM3 | REFS | 24 | | | | | | |
| 0 | NTRI | INTEGER | | C017 | REFS | 15 | 38 | 69 | DEFINED | 38 | 69 | |
| 2 | NTR1 | INTEGER | | PHILM3 | REFS | 24 | | | | | | |
| 1 | NTR2 | INTEGER | | PHILM3 | REFS | 24 | | | | | | |
| 0 | NTR3 | INTEGER | | PHILM3 | REFS | 24 | 103 | 109 | 128 | DEFINED | 103 | 109 |
| | | | | | 128 | | | | | | | |
| 5 | NWORK | INTEGER | | C015 | REFS | 14 | 35 | 37 | 38 | 39 | 65 | 68 |
| | | | | | 69 | 70 | 2*130 | DEFINED | 35 | 65 | | |
| 0 | PSMC | REAL | | C045 | REFS | 22 | 94 | | | | | |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | PSUC | REAL | PERM | REFS | 23 | | | | | | |
| 2261 | PTR | INTEGER | COM1 | REFS | 8 | 25 | 46 | | | | |
| 0 | RATINT | REAL | C031 | REFS | 19 | | | | | | |
| 0 | REASON | INTEGER | COM7 | REFS | 10 | 25 | DEFINED | 126 | | | |
| 2264 | RECOV | REAL | COM1 | REFS | 8 | 123 | | | | | |
| 5 | RMC | REAL | COM9 | REFS | 12 | 31 | | | | | |
| 1 | TABLE | REAL | ARRAY COM1 | REFS | 8 | 46 | | | | | |
| 76 | TC | REAL | COM3 | REFS | 9 | | | | | | |
| 2266 | TIME | REAL | COM1 | REFS | 8 | 31 | 46 | 54 | 2*86 | 114 | 117 |
| | | | | | 121 | 122 | 2*123 | DEFINED | 66 | 114 | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| DETTIM | | 3 | 117 |
| FIFAU | | 3 | 47 |
| GATHER | | 0 | 112 |
| MISCYC | | 3 | 123 |
| RANF | REAL | 1 | 94 |

| INLINE FUNCTIONS | TYPE | ARGS | DEF LINE | REFERENCES |
|---|---|---|---|---|
| INT | INTEGER | 1 INTRIN | | 77 |

| STATEMENT LABELS | | DEF LINE | REFERENCES | | | | |
|---|---|---|---|---|---|---|---|
| 41 | 10 | 44 | 31 | 49 | | | |
| 55 | 20 | 50 | 46 | | | | |
| 75 | 21 | 61 | 55 | 57 | 58 | | |
| 77 | 22 | 62 | 56 | | | | |
| 125 | 23 | 72 | 63 | | | | |
| 36 | 24 | 41 | 34 | | | | |
| 0 | 40 | 78 | 74 | | | | |
| 144 | 50 | 79 | 75 | | | | |
| 230 | 60 | 110 | 82 | 88 | 92 | | |
| 233 | 61 | 111 | 83 | 94 | | | |
| 171 | 70 | 93 | 86 | | | | |
| 160 | 90 | 89 | 95 | | | | |
| 0 | 100 | 119 | 115 | | | | |
| 256 | 110 | 120 | 116 | | | | |
| 222 | 1110 | 105 | 98 | 99 | 100 | 101 | |
| 224 | 1111 | 106 | 97 | 104 | | | |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | | | |
|---|---|---|---|---|---|---|---|---|
| 65 | 21 | * I | 55 61 | 128 | OPT | EXITS | | |
| 133 | 40 | * I | 74 78 | 118 | OPT | EXITS | | |
| 146 | 60 | * J | 82 110 | 658 | EXT REFS | EXITS | NOT INNER | |
| 210 | 1110 | * I | 98 105 | 148 | OPT | EXITS | | |
| 243 | 100 | * J | 115 119 | 138 | EXT REFS | EXITS | | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | |
|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 IDIM (1) | 1 TABLE (1200) | 1201 PTR (1) | | |
| | | 1202 EXTEN (1) | 1203 IDETEC (1) | 1204 RECOV (1) | | |
| | | 1205 DELAY (1) | 1206 TIME (1) | | | |
| COM3 | 63 | 0 ACFAU (60) | 60 ENDMIS (1) | 61 MEMSIZ (1) | | |
| | | 62 TC (1) | | | | |
| COM7 | 1 | 0 REASON (1) | | | | |
| COM8 | 1 | 0 FOCO (1) | | | | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| COH9 | 7 | 0 | COPLAS | (5) | 5 | RHC | (1) | 6 | DURHC | (1) |
| C011 | 1 | 0 | MISTAK | (1) | | | | | | |
| C015 | 6 | 0 | NOON | (5) | 5 | NWORK | (1) | | | |
| C017 | 2 | 0 | NTRI | (1) | 1 | NQUA | (1) | | | |
| C018 | 1 | 0 | IREP | (1) | | | | | | |
| C028 | 2 | 0 | MODSIM | (1) | 1 | NSPB | (1) | | | |
| C029 | 1 | 0 | NMC | (1) | | | | | | |
| C031 | 2 | 0 | RATINT | (1) | 1 | ISYNC | (1) | | | |
| C041 | 2 | 0 | ICATAS | (1) | 1 | I3 | (1) | | | |
| C042 | 3 | 0 | MISTKI | (3) | | | | | | |
| C045 | 1 | 0 | PSMC | (1) | | | | | | |
| PERM | 7 | 0 | LAST | (5) | 5 | MININT | (1) | 6 | PSUC | (1) |
| PHILM3 | 4 | 0 | NTR3 | (1) | 1 | NTR2 | (1) | 2 | NTR1 | (1) |
| | | 3 | NTNR1 | (1) | | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 343B | 227 |
| COMMON LENGTH | 2437B | 1311 |

```
      SUBROUTINE STATE8(NEXT)
C                        THIS VERSION:   MARCH 1976
C THIS IS A ROLLBACK
C
C ***************************************************************************
C       IREPET INDICATES A SUCCESSFUL ROLLBACK IF 0 .                      OL29APR4
C       WHEN ROLLBACK IS SUCCESSFUL WE HAVE TO DETERMINE WHICH COMPUTEROL29APR4
C       IS STILL GOOD, THIS IS DONE IN STATE9                              OL29APR4
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/COM7/REASON
      COMMON/CO10/MAXRLB,DURRA
      COMMON/CO16/NDIAG,NUNDI
      COMMON/CO25/END
      COMMON/CO31/RATINT,ISYNC
      COMMON/CO32/IBAD
      COMMON/CO50/IMOBA
      COMMON/COVER/ITLKP,IFNE                                          SP11APR4
      COMMON/PHILM3/NTR3,NTR2,NTR1,NTNR1
      COMMON/PBACK/RBCPU
      INTEGER EXTEN,PTR,REASON
C********** ********* ********* ********** ********* ********* ********
C
C       SET DETECTION TIME OF FAULT CAUSING THIS ROLLBACK                SP09APR4
      IDELET=0
      END=ACFAU(IDETEC,2)
      IMOBA=ACFAU(IDETEC,3)
      IBAD=ACFAU(IDETEC,4)
      TIME=ACFAU(IDETEC,6)
      DURRB=DURRA
      IF (ISYNC.EQ.0) DURRB=DURRA*2.*RANF(0.)
      TIMI=TIME
      LRB=0
      IF(MAXRLB.GT.0) GOTO 10
      NEXT=9
      RETURN
   10 CONTINUE
      LRB=LRB+1
   40 CONTINUE
C       SEE IF NEXT FAULT OCCURANCE IS AFTER THE TIME IT WILL BE AFTER SP12APR4
C       THE ROLLBACK IS COMPLETE                                         SP09APR4
C
      IF (TABLE(PTR,1).GT.TIME+DURRB) GO TO 20
C
C       SEE IF WE ARE POINTING TO A FAULT IN A LIVE COMPUTER              P09APR4
      CALL FIFAU(IN,NEXT,1)
      IF(NEXT.EQ.5) RETURN
      GO TO 40
   20 CONTINUE
C       NICE CLEAN BREAK BETWEEN FAULTS                                  SP09APR4
      IREPET=0
      IF (ISYNC.EQ.1) GO TO 32
      DO  30 J=1,IDIM
      IF (ACFAU(J,3).EQ.0.) GO TO 32
      JJ=J
```

```
                30 CONTINUE
                32 CONTINUE
                   IEFFA=0
                   DO 80 J=1,IDIM
60                 ACF3=ACFAU(J,3)
                   IF (ACF3.EQ.0.) GO TO 81
                   IF (ACFAU(J,6).GT.TIME) GO TO 90
             C        SEE IF THERE IS AN ACTIVE FAULT THAT DISAPPEARS AFTER THE      SP09APR4
             C        DETECTION OF THIS FAULT (THAT HAS BEEN DETECTED)               SP09APR4
65                 IF (ACFAU(J,2).GT.TIME) GO TO 100
             C        SEE IF THERE IS AN EXTENT ASSOCIATED WITH THE FAULT (MEANS     SP09APR4
             C        THE ROLLBACK WON T HELP FIX IT)                                SP09APR4
                   IF (ACFAU(J,5).NE.0.) GO TO 100
                   IF(RANF(0.0).GE.RBCPU) GOTO 100
70                 IF (ISYNC.EQ.1) GO TO 1111
                   IDELET=1
                   DO  1110 I=1,JJ
                   IF (ACFAU(I,3).EQ.0.) GO TO 1110
                   IF (I.EQ.J) GO TO 1110
75                 IF (ACFAU(I,2).NE.ACFAU(J,2)) GO TO 1110
                   IDELET=IDELET-1
                   NTR2=NTR2-1
                   GO TO 1111
                1110 CONTINUE
80              1111 CONTINUE
                   IEFFA=1
                   ACFAU(J,3)=0.
                   NTR2 = NTR2 + 1                                                   OL29APR4
                   GO TO 80
85           C
                90 CONTINUE
             C A ROLLBACK MAY DELAY THE DETECTION OF A LURKING FAULT
                   ACFAU(J,6)=ACFAU(J,6)+DURRB
                   GO TO 80
90           C
               100 CONTINUE
             C
                   IREPET=1
                80 CONTINUE
95              81 CONTINUE
                   TIME=TIME+DURRB
                   IF (IEFFA.EQ.1) CALL GATHER
             C
             C
100          C        HOT DAMN -- WE RE STILL IN BUSINESS, THE ROLLBACK HAS          SP09APR4
             C        SUCCESSFUL                                                     SP09APR4
                   NEXT = 4                                                          SP09APR4
                   CALL MISCYC(TIMI,TIME,NEXT)
                   IF (IREPET.EQ.1) GO TO 160
105                IF (NEXT.NE.5)RETURN
                   GO TO 161
             C
               160 CONTINUE
             C
110          C        SEE IF WE TRY THE ROLLBACK AGAIN                               SP09APR4
```

```
      IF(NEXT.EQ.5) GO TO 161
      IF (LRB.LT.MAXRLB) GO TO 10
C                                                                  OL29APR4
      NEXT = 9
115   RETURN
  161 CONTINUE
      NDIAG=NDIAG+1
      NUNDI=NUNDI+1
      IF (IDELET.EQ.1) NTR2=NTR2-1
120   IF (END.LT.ENDMIS) ITLKP=ITLKP+1
      REASON=6
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | DEF LINE | REFERENCES | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 STATE8 | 1 | 36 | 47 | 105 | 115 | 122 | | | | | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 ACFAU | | REAL | ARRAY | COM3 | REFS | 10 | 26 | 27 | 28 | 29 | 54 | 60 |
| | | | | | 62 | 65 | 68 | 73 | 2*75 | 88 | | |
| | | | | | DEFINED | 82 | 88 | | | | | |
| 242 ACF3 | | REAL | | | REFS | 61 | DEFINED | 60 | | | | |
| 2265 DELAY | | REAL | | COM1 | REFS | 9 | | | | | | |
| 1 DURRA | | REAL | | C010 | REFS | 12 | 30 | 31 | | | | |
| 232 DURRB | | REAL | | | REFS | 43 | 88 | 96 | DEFINED | 30 | 31 | |
| 0 END | | REAL | | C025 | REFS | 14 | 120 | DEFINED | 26 | | | |
| 74 ENDMIS | | REAL | | COM3 | REFS | 10 | 120 | | | | | |
| 2262 EXTEN | | INTEGER | | COM1 | REFS | 9 | 21 | | | | | |
| 243 I | | INTEGER | | | REFS | 73 | 74 | 75 | DEFINED | 72 | | |
| 0 IBAD | | INTEGER | | C032 | REFS | 16 | DEFINED | 28 | | | | |
| 231 IDELET | | INTEGER | | | REFS | 76 | 119 | DEFINED | 25 | 71 | 76 | |
| 2263 IDETEC | | INTEGER | | COM1 | REFS | 9 | 26 | 27 | 28 | 29 | | |
| 0 IDIM | | INTEGER | | COM1 | REFS | 9 | 53 | 59 | | | | |
| 241 IEFFA | | INTEGER | | | REFS | 97 | DEFINED | 58 | 81 | | | |
| 1 IFWE | | INTEGER | | COVER | REFS | 18 | | | | | | |
| 0 IMOBA | | INTEGER | | C050 | REFS | 17 | DEFINED | 27 | | | | |
| 235 IN | * | INTEGER | | | REFS | 46 | | | | | | |
| 236 IREPET | | INTEGER | | | REFS | 104 | DEFINED | 51 | 93 | | | |
| 1 ISYNC | | INTEGER | | C031 | REFS | 15 | 31 | 52 | 70 | | | |
| 0 ITLKP | | INTEGER | | COVER | REFS | 18 | 120 | DEFINED | 120 | | | |
| 237 J | | INTEGER | | | REFS | 54 | 55 | 60 | 62 | 65 | 68 | 74 |
| | | | | | 75 | 82 | 2*88 | DEFINED | 53 | 59 | | |
| 240 JJ | | INTEGER | | | REFS | 72 | DEFINED | 55 | | | | |
| 234 LRB | | INTEGER | | | REFS | 38 | 112 | DEFINED | 33 | 38 | | |
| 0 MAXRLB | | INTEGER | | C010 | REFS | 12 | 34 | 112 | | | | |
| 75 MEMSIZ | | INTEGER | | COM3 | REFS | 10 | | | | | | |
| 0 NDIAG | | INTEGER | | C016 | REFS | 13 | 117 | DEFINED | 117 | | | |
| 0 NEXT | | INTEGER | | F.P. | REFS | 46 | 47 | 103 | 105 | 111 | | |
| | | | | | DEFINED | 1 | 35 | 102 | 114 | | | |
| 3 NTNR1 | | INTEGER | | PHILM3 | REFS | 19 | | | | | | |
| 2 NTR1 | | INTEGER | | PHILM3 | REFS | 19 | | | | | | |
| 1 NTR2 | | INTEGER | | PHILM3 | REFS | 19 | 77 | 83 | 119 | DEFINED | 77 | 83 |
| | | | | | 119 | | | | | | | |
| 0 NTR3 | | INTEGER | | PHILM3 | REFS | 19 | | | | | | |
| 1 NUNDI | | INTEGER | | C016 | REFS | 13 | 118 | DEFINED | 118 | | | |
| 2261 PTR | | INTEGER | | COM1 | REFS | 9 | 21 | 43 | | | | |
| 0 RATINT | | REAL | | C031 | REFS | 15 | | | | | | |
| 0 RBCPU | | REAL | | RBACK | REFS | 20 | 69 | | | | | |
| 0 REASON | | INTEGER | | COM7 | REFS | 11 | 21 | DEFINED | 121 | | | |
| 2264 RECOV | | REAL | | COM1 | REFS | 9 | | | | | | |
| 1 TABLE | | REAL | ARRAY | COM1 | REFS | 9 | 43 | | | | | |
| 76 TC | | REAL | | COM3 | REFS | 10 | | | | | | |
| 2266 TIME | | REAL | | COM1 | REFS | 9 | 32 | 43 | 62 | 65 | 96 | 103 |
| | | | | | DEFINED | 29 | 96 | | | | | |
| 233 TIMI | | REAL | | | REFS | 103 | DEFINED | 32 | | | | |

```
EXTERNALS          TYPE   ARGS      REFERENCES
       FIFAU                 3           46
       GATHER                0           97
       MISCYC                3          103
       RANF       REAL       1           31        69

STATEMENT LABELS              DEF LINE    REFERENCES
   30  10                        37          34        112
   46  20                        49          43
    0  30                        56          53
   62  32                        57          52        54
   32  40                        39          48
  141  80                        94          59        84        89
  144  81                        95          61
  135  90                        86          62
  140  100                       91          65        68        69
  166  160                      108         104
  176  161                      116         106       111
  127  1110                      79          72        73        74        75
  131  1111                      80          70        78

LOOPS  LABEL     INDEX     FROM-TO    LENGTH    PROPERTIES
   56  30      * J           53 56        48     INSTACK    EXITS
   64  80      * J           59 94       608              EXT REFS    EXITS   NOT INNER
  115  1110    * I           72 79       148         OPT     EXITS

COMMON BLOCKS    LENGTH     MEMBERS - BIAS NAME(LENGTH)
       COM1        1207              0 IDIM   (1)              1 TABLE  (1200)          1201 PTR    (1)
                                  1202 EXTEN  (1)           1203 IDETEC (1)           1204 RECOV  (1)
                                  1205 DELAY  (1)           1206 TIME   (1)
       COM3          63            0 ACFAU  (60)             60 ENDMIS (1)             61 MEMSIZ (1)
                                    62 TC     (1)
       COM7           1            0 REASON (1)
       CO10           2            0 MAXRLB (1)              1 DURRA  (1)
       CO16           2            0 NDIAG  (1)              1 NUNDI  (1)
       CO25           1            0 END    (1)
       CO31           2            0 RATINT (1)              1 ISYNC  (1)
       CO32           1            0 IBAD   (1)
       CO50           1            0 IMOBA  (1)
       COVER          2            0 ITLKP  (1)              1 IFWE   (1)
       PHILM3         4            0 NTR3   (1)              1 NTR2   (1)              2 NTR1   (1)
                                    3 NTNR1  (1)
       RBACK          1            0 RBCPU  (1)

STATISTICS
  PROGRAM LENGTH    2448     164
   COMMON LENGTH    2407B   1287
```

```
      SUBROUTINE STATE9(NEXT)                                          SP09APR4
C                    THIS VERSION:  MARCH 1976
C       THIS IS THE FAMOUS DIAGONISTIC AND RECOVERY STATE.
C       THE ONLY WAY TO ARRIVE AT THIS STATE IS TO HAVE HAD THE DUPLEX
C       MODE RECOVERY TECHNIQUE (NAMELY ROLLBACK) NOT SUCCESSFUL ENOUGH
C       (THIS HINGES ON THE PROBABILITY OF SUCCESS OF A ROLLBACK
C
C********** ********* ********* ********* ********* ********* ********
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC                         SP29APR4
      COMMON/COM7/REASON                                               SP08APR4
      COMMON/CO15/NOON(5),NWORK
      COMMON/CO16/NDIAG,NUNDI
      COMMON/CO25/END
      COMMON/CO26/TW2
      COMMON/CO32/IRAD
      COMMON/CO50/IMOBA
      COMMON/COVER/ITLKP,IFHE
      COMMON/D/DIAGN
      COMMON/DETE/P,DETMAX,PDM
      COMMON/FAILD2/FOCO2                                              SP29APR4
      INTEGER  PTR,EXTEN                                               SP29APR4
      INTEGER  REASON                                                  SP10APR4
C ****************************************************************
C
C A TRANSIENT FAULT CANNOT BE DIAGNOSED IF IT HAS DISAPPEARED
      PDET=PDM
      IF(IMOBA.EQ.1) PDET=P
   68 U=RANF(0.)
      IF (U.LT.PDET) GO TO 169
      IF (END.LT.TIME) GO TO 110
      IF (U.GT.DETMAX) GO TO 101
      U=DIAGN*2.*(U-PDET)/(DETMAX-PDET)
  171 CONTINUE
      CALL  HISCYC(TIME,TIME+U+TW2,NEXT)
      TIME=TIME+U+TW2
      IF (NEXT.NE.5) GO TO 100
  101 CONTINUE
      NEXT=5
      REASON=4
C        TOO BAD ABOUT THAT -- IT S A SYSTEM FAILURE                   SP08APR4
      NUNDI=NUNDI+1
      IF (END.LT.ENDMIS) ITLKP=ITLKP+1
      RETURN                                                           SP08APR4
  110 CONTINUE
      IF (RANF(0.).GT.0.5) GO TO 101
      U=2.*DIAGN
      GO TO 171
C                                                                      SP08APR4
  100 CONTINUE                                                         SP08APR4
C        DETERMINE THE ONLY GOOD COMPUTER                              SP29APR4
      DO 150 J=1,5
      IF ((NOON(J).EQ.0). OR.(IRAD.EQ.J)) GO TO  150
      FOCO2=J
      GO TO 200
```

```
      150 CONTINUE
      200 CONTINUE
          NOON(IBAD)=0
          NHORK=1
      201 CONTINUE
          IF(TABLE(PTR,1).GE.TIME) GO TO 210
          CALL FIFAU(IN,NEXT,1)
          IF(NEXT.EQ.5) RETURN
          IF(IN.EQ.1) GOTO 101
          GOTO 201
      210 CONTINUE
C
C        HELL, YOU RE NOT A WHOLE LOT BETTER OFF, BUT AT LEAST A SIMPLEXSP08APR4
C        COMPUTER SYSTEM MIGHT KEEP THE PLANE IN THE AIR.              SP08APR4
          NEXT = 10                                                    SP08APR4
          RETURN                                                       SP08APR4
      169 CONTINUE
          U=0.
          GO TO 171
          END                                                          SP08APR4
```

SYMBOLIC REFERENCE MAP

```
ENTRY POINTS     DEF LINE     REFERENCES
    2  STATE9          1        44        63        71

VARIABLES     SN  TYPE           RELOCATION
    0  ACFAU      REAL       ARRAY    COM3          REFS       10
 2265  DELAY      REAL                COM1          REFS        9
    1  DETMAX     REAL                DETE          REFS       20        32        33
    0  DIAGN      REAL                0             REFS       19        33        47
    0  END        REAL                C025          REFS       14        31        43
   74  ENDMIS     REAL                COM3          REFS       10        43
 2262  EXTEN      INTEGER             COM1          REFS        9        22
    0  FOCO2      REAL                FAILD2        REFS       21   DEFINED        54
    0  IBAD       INTEGER             C032          REFS       16        53        58
 2263  IDETEC     INTEGER             COM1          REFS        9
    0  IDIM       INTEGER             COM1          REFS        9
    1  IFWE       INTEGER             COVER         REFS       18
    0  IMOBA      INTEGER             C050          REFS       17        28
  147  IN         INTEGER                           REFS       62        64
    0  ITLKP      INTEGER             COVER         REFS       18        43   DEFINED        43
  146  J          INTEGER                           REFS     2*53        54   DEFINED        52
   75  MEMSIZ     INTEGER             COM3          REFS       10
    0  NDIAG      INTEGER             C016          REFS       13
    0  NEXT       INTEGER              F.P.         REFS       35        37        62        63   DEFINED        39
                                                     70
    0  NOON       INTEGER    ARRAY    C015          REFS       12        53   DEFINED        58
    1  NUNDI      INTEGER             C016          REFS       13        42   DEFINED        42
    5  NWORK      INTEGER             C015          REFS       12   DEFINED        59
    0  P          REAL                DETE          REFS       20        28
  144  PDET       REAL                              REFS       30      2*33   DEFINED        27        28
    2  PDM        REAL                DETE          REFS       20        27
 2261  PTR        INTEGER             COM1          REFS        9        22        61
    0  REASON     INTEGER             COM7          REFS       11        23   DEFINED        40
 2264  RECOV      REAL                COM1          REFS        9
    1  TABLE      REAL       ARRAY    COM1          REFS        9        61
   76  TC         REAL                COM3          REFS       10
 2266  TIME       REAL                COM1          REFS        9        31      2*35        36        61
                                                  DEFINED       36
    0  TH2        REAL                C026          REFS       15        35        36
  145  U          REAL                              REFS       30        32        33        35        36
                                                  DEFINED       29        33        47        73

EXTERNALS          TYPE   ARGS       REFERENCES
       FIFAU                 3            62
       MISCYC                3            35
       RANF       REAL       1            29        46

STATEMENT LABELS              DEF LINE    REFERENCES
    0  68          INACTIVE       29
   62  100                        50        37
   43  101                        38        32        46        64
   54  110                        45        31
   74  150                        56        52        53
```

| STATEMENT LABELS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|
| 123 | 169 | 72 | 30 | |
| 27 | 171 | 34 | 48 | 74 |
| 76 | 200 | 57 | 55 | |
| 101 | 201 | 60 | 65 | |
| 120 | 210 | 66 | 61 | |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | EXITS |
|---|---|---|---|---|---|---|
| 66 | 150 | * J | 52 56 | 100 | OPT | |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS | NAME | (LENGTH) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 | IDIM | (1) | 1 | TABLE | (1200) | 1201 | PTR | (1) | |
| | | 1202 | FXTEN | (1) | 1203 | IDETEC | (1) | 1204 | RECOV | (1) | |
| | | 1205 | DELAY | (1) | 1206 | TIME | (1) | | | | |
| COM3 | 63 | 0 | ACFAU | (60) | 60 | ENDMIS | (1) | 61 | MEMSIZ | (1) | |
| | | 62 | TC | (1) | | | | | | | |
| COM7 | 1 | 0 | REASON | (1) | | | | | | | |
| CO15 | 6 | 0 | NOON | (5) | 5 | NWORK | (1) | | | | |
| CO16 | 2 | 0 | NDIAG | (1) | 1 | NUNDI | (1) | | | | |
| CO25 | 1 | 0 | END | (1) | | | | | | | |
| CO26 | 1 | 0 | TW2 | (1) | | | | | | | |
| CO32 | 1 | 0 | IBAD | (1) | | | | | | | |
| CO50 | 1 | 0 | IMOBA | (1) | | | | | | | |
| COVER | 2 | 0 | ITLKP | (1) | 1 | IFWE | (1) | | | | |
| D | 1 | 0 | DIAGN | (1) | | | | | | | |
| DETF | 3 | 0 | P | (1) | 1 | DETMAX | (1) | 2 | PDM | (1) | |
| FAIL02 | 1 | 0 | FOC02 | (1) | | | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 1508 | 104 |
| COMMON LENGTH | 24128 | 1290 |

```
      SUBROUTINE STATEA(NEXT)                                          SP08APR4
C                     THIS VERSION:  MARCH 1976
C        THIS IS THE PRECARIOUSLY POSTURED SIMPLEX COMPUTER SYSTEM STATESP08APR4
C        IN THIS STATE YOU VE REALLY GOT YOURSELF OUT ON A LIMB BECAUSE SP08APR4
C        THERE S ABSOLUTELY NO REDUNDANCY - YOU RE ZINGING IN THERE ON  SP08APR4
C        ONE AND ONLY ONE COMPUTER                                      SP08APR4
C ********* ********* ********* ********** ********** ********** *****  SP08APR4
C
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC                         SP09APR4
      COMMON/COM7/REASON
      COMMON/C031/RATINT,ISYNC
      COMMON/C060/LSTFLT
      COMMON/COVER/ITLKP,IFWE                                          SP29APR4
      COMMON/FAILD2/FOCO2                                              SP08APR4
      INTEGER PTR,EXTEN,REASON
C ***************************************************************
C
C SUPPRESSION OF THE FAULTS IN THE SWITCHED OFF COMPUTER
      IRA=0
      DO 10 J=1,IDIM
      IF (ACFAU(J,3).EQ.0.)GO TO 20
      IF (ACFAU(J,4).EQ.FOCO2) GO TO 10
      ACFAU(J,3)=0.
      IRA=IRA+1
      IF ((ACFAU(J,2).LT.ENDMIS).AND.(IRA.EQ.1)) ITLKP=ITLKP+1
   10 CONTINUE
   20 CONTINUE
      IF (IRA.GE.1) CALL GATHER
C IF THERE IS A LURKING FAULT GO TO 30
      IF (ACFAU(1,3).NE.0.)GO TO 30
C IF THERE IS NO MORE FAULT, RETURN
   70 CONTINUE
      TIME=TABLE(PTR,1)
      IF (TIME.LT.ENDMIS) GO TO 40
      NEXT=6
      RETURN
   40 CONTINUE
      CALL FIFAU(IN,NEXT,1)
      IF(NEXT.EQ.5)RETURN
      IF(IN.EQ.0) GOTO 70
C IN STATEB, WHEN ACFAU(J,6) IS -1, IT IS DETERMINED IF THE FAULT IS
C DETECTED OR NOT. IF WE FOLLOWED EXACTLY THE STATE DIAGRAM, IT SHOULD
C BE DETERMINED HERE.
      ACFAU(LSTFLT,6)=-1
      PTR=PTR+1
      NEXT=11
      RETURN
   30 CONTINUE
      TIMI=ACFAU(1,1)
      DO 1 J=2,IDIM
      IF (ACFAU(J,3).EQ.0.) GO TO 2
      TIMI=AMIN1(TIMI,ACFAU(J,2))
    1 CONTINUE
    2 CONTINUE
```

```
      TIME=AMAX1(TIMI,TIME)
      NEXT=11
      END                                                    SP08APR4
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | | |
|---|---|---|---|---|---|---|
| 2 | STATEA | 1 | 37 | 40 | 48 | 58 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 10 | 22 | 23 | 26 | 31 | 50 | 52 |
| | | | | | 53 | DEFINED | 24 | 45 | | | | |
| 2265 | DELAY | REAL | | COM1 | REFS | 9 | | | | | | |
| 74 | ENDMIS | REAL | | COM3 | REFS | 10 | 26 | 35 | | | | |
| 2262 | EXTEN | INTEGER | | COM1 | REFS | 9 | 16 | | | | | |
| 0 | FOCO2 | REAL | | FAILD2 | REFS | 15 | 23 | | | | | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 9 | | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 9 | 21 | 51 | | | | |
| 1 | IFWE | INTEGER | | COVER | REFS | 14 | | | | | | |
| 115 | IN | INTEGER | | | REFS | 39 | 41 | | | | | |
| 113 | IRA | INTEGER | | | REFS | 25 | 26 | 29 | DEFINED | 20 | 25 | |
| 1 | ISYNC | INTEGER | | C031 | REFS | 12 | | | | | | |
| 0 | ITLKP | INTEGER | | COVER | REFS | 14 | 26 | DEFINED | 26 | | | |
| 114 | J | INTEGER | | | REFS | 22 | 23 | 24 | 26 | 52 | 53 | |
| | | | | | DEFINED | 21 | 51 | | | | | |
| 0 | LSTFLT | INTEGER | | C060 | REFS | 13 | 45 | | | | | |
| 75 | MEMSIZ | INTEGER | | COM3 | REFS | 10 | | | | | | |
| 0 | NEXT | INTEGER | | F.P. | REFS | 39 | 40 | DEFINED | 1 | 36 | 47 | 57 |
| 2261 | PTR | INTEGER | | COM1 | REFS | 9 | 16 | 34 | 46 | DEFINED | 46 | |
| 0 | RATINT | REAL | | C031 | REFS | 12 | | | | | | |
| 0 | REASON | INTEGER | | COM7 | REFS | 11 | 16 | | | | | |
| 2264 | RECOV | REAL | | COM1 | REFS | 9 | | | | | | |
| 1 | TABLE | REAL | ARPAY | COM1 | REFS | 9 | 34 | | | | | |
| 76 | TC | REAL | | COM3 | REFS | 10 | | | | | | |
| 2266 | TIME | REAL | | COM1 | REFS | 9 | 35 | 56 | DEFINED | 34 | 56 | |
| 116 | TIMI | REAL | | | REFS | 53 | 56 | DEFINED | 50 | 53 | | |

| EXTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| FIFAU | | 3 | 39 |
| GATHER | | 0 | 29 |

| INLINE FUNCTIONS | TYPE | ARGS | DEF LINE | REFERENCES |
|---|---|---|---|---|
| AMAX1 | REAL | 0 | INTRIN | 56 |
| AMIN1 | REAL | 0 | INTRIN | 53 |

| STATEMENT LABELS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|
| 0 | 1 | 54 | 51 | |
| 100 | 2 | 55 | 52 | |
| 25 | 10 | 27 | 21 | 23 |
| 27 | 20 | 28 | 22 | |
| 64 | 30 | 49 | 31 | |
| 44 | 40 | 38 | 35 | |
| 36 | 70 | 33 | 41 | |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | |
|---|---|---|---|---|---|---|
| 12 | 10 | * J | 21 27 | 158 | OPT | EXITS |
| 72 | 1 | * J | 51 54 | 68 | INSTACK | EXITS |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS | NAME | (LENGTH) |
|---|---|---|---|---|
| COM1 | 1207 | 0 | IDIM | (1) |
| | | 1 | TABLE | (1200) |
| | | 1201 | PTR | (1) |
| | | 1202 | FXTEN | (1) |
| | | 1203 | IDETEC | (1) |
| | | 1204 | RECOV | (1) |
| | | 1205 | DELAY | (1) |
| | | 1206 | TIME | (1) |
| COM3 | 63 | 0 | ACFAU | (60) |
| | | 60 | ENDMIS | (1) |
| | | 61 | MEMSIZ | (1) |
| | | 62 | TC | (1) |
| COM7 | 1 | 0 | REASON | (1) |
| CO31 | 2 | 0 | RATINT | (1) |
| | | 1 | ISYNC | (1) |
| CO60 | 1 | 0 | LSTFLT | (1) |
| COVER | 2 | 0 | ITLKP | (1) |
| | | 1 | IFHE | (1) |
| FAILD2 | 1 | 0 | FOCO2 | (1) |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 1178 | 79 |
| COMMON LENGTH | 23750 | 1277 |

```
      SUBROUTINE STATEB(NEXT)
C                        THIS VERSION:   MARCH 1976
C THIS CORRESPONDS TO  DETECTION AND ROLLBACK IN SIMPLEX
C
C ***************************************************************************
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/COM7/REASON
      COMMON/CO10/M,DURRA
      COMMON/CO31/RATINT,ISYNC
      COMMON/CO60/LSTFLT
      COMMON/DETE/PDET,DETMAX,PDM
      COMMON/FAILD2/FOCO2
      COMMON/PHILM3/NTR3,NTR2,NTR1,NTNR1
      INTEGER PTR,EXTEN
      INTEGER REASON
C ***************************************************************************
C
      TIMI=TIME
      IF (ISYNC.EQ.0) GO TO 1
      DURRB=TIME-AINT(TIME/TC)*TC
      GO TO 40
    1 CONTINUE
      DURRB=DURRA*2.*RANF(0.)
   40 CONTINUE
      IF (TABLE(PTR,1).GT.TIME+DURRB) GO TO 20
      CALL FIFAU(IN,NEXT,1)
      IF(NEXT.EQ.5) RETURN
      IF(IN.EQ.1) ACFAU(LSTFLT,6)=-1
      GO TO 40
   20 CONTINUE
      IREPET=0
      IEFFA=0
      DO 80 J=1,IDIM
      ACF3=ACFAU(J,3)
      IF (ACF3.EQ.0.) GO TO 81
      IF (ACFAU(J,6).EQ.0.)GO TO 82
C TEST ON THE DETECTION OF THIS FAULT
      U=RANF(0.)
      IF(((ACF3.EQ.3.).AND.(U.LT.PDM)).OR.
     1   ((ACF3.EQ.1.).AND.(U.LT.PDET))) GO TO 83
      IF (ACFAU(J,2).LT.ENDMIS) NTNR1=NTNR1+1
      NEXT=5
      REASON=5
      RETURN
   83 CONTINUE
      ACFAU(J,6)=0.
   82 CONTINUE
      IF (ACFAU(J,2).GT.TIME) GO TO 100
      IF (ACFAU(J,5).NE.0.) GO TO 100
      NTR1=NTR1+1
      IEFFA=1
      ACFAU(J,3)=0.
      GO TO 80
  100 CONTINUE
```

```
                IREPET=1
             80 CONTINUE
             81 CONTINUE
                TIME=TIME+DURRB
60              IF (IEFFA.EQ.1) CALL GATHER
                REASON=5
                CALL MISCYC(TIMI,TIME,NEXT)
                IF (NEXT.EQ.5) GO TO 10
                IF (IREPET.EQ.1) GO TO 40
65              NEXT=10
                RETURN
             10 CONTINUE
                DO 50 I=1,IDIM
                IF (ACFAU(I,3).EQ.0.) RETURN
70              IF (ACFAU(I,2).LT.ENDHIS)NTNR1=NTNR1+1
             50 CONTINUE
                END
```

SYMBOLIC REFERENCE MAP.

| ENTRY POINTS | DEF LINE | REFERENCES | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2 STATEB | 1 | 28 | 45 | 66 | 69 | 72 | | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 7 | 35 | 37 | 42 | 49 | 50 | 69 |
| | | | | | 70 | DEFINED | 29 | 47 | 53 | | | |
| 210 | ACF3 | REAL | | | REFS | 36 | 2*40 | DEFINED | 35 | | | |
| 2265 | DELAY | REAL | | COM1 | REFS | 6 | | | | | | |
| 1 | DETMAX | REAL | | DETE | REFS | 12 | | | | | | |
| 1 | DURRA | REAL | | C010 | REFS | 9 | 24 | | | | | |
| 203 | DURRB | REAL | | | REFS | 26 | 59 | DEFINED | 21 | 24 | | |
| 74 | ENDMIS | REAL | | COM7 | REFS | 7 | 42 | 70 | | | | |
| 2262 | EXTFN | INTEGER | | COM1 | REFS | 6 | 15 | | | | | |
| 0 | FOCO2 | REAL | | FAILO2 | REFS | 13 | | | | | | |
| 212 | I | INTEGER | | | REFS | 69 | 70 | DEFINED | 68 | | | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 6 | | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 6 | 34 | 68 | | | | |
| 206 | IEFFA | INTEGER | | | REFS | 60 | DEFINED | 33 | 52 | | | |
| 204 | IN | INTEGER | | | REFS | 27 | 29 | | | | | |
| 205 | IREPET | INTEGER | | | REFS | 64 | DEFINED | 32 | 56 | | | |
| 1 | ISYNC | INTEGER | | C031 | REFS | 10 | 20 | | | | | |
| 207 | J | INTEGER | | | REFS | 35 | 37 | 42 | 47 | 49 | 50 | 53 |
| | | | | | DEFINED | 34 | | | | | | |
| 0 | LSTFLT | INTEGER | | C060 | REFS | 11 | 29 | | | | | |
| 0 | M | INTEGER | | C010 | REFS | 9 | | | | | | |
| 75 | MEMSIZ | INTEGER | | COM3 | REFS | 7 | | | | | | |
| 0 | NEXT | INTEGER | | F.P. | REFS | 27 | 28 | 62 | 63 | DEFINED | 1 | 43 |
| | | | | | 65 | | | | | | | |
| 3 | NTNR1 | INTEGER | | PHILM3 | REFS | 14 | 42 | 70 | DEFINED | 42 | 70 | |
| 2 | NTR1 | INTEGER | | PHILM3 | REFS | 14 | 51 | DEFINED | 51 | | | |
| 1 | NTR2 | INTEGER | | PHILM3 | REFS | 14 | | | | | | |
| 0 | NTR3 | INTEGER | | PHILM3 | REFS | 14 | | | | | | |
| 0 | PDET | REAL | | DETE | REFS | 12 | 40 | | | | | |
| 2 | PDM | REAL | | DETE | REFS | 12 | 40 | | | | | |
| 2261 | PTR | INTEGER | | COM1 | REFS | 6 | 15 | 26 | | | | |
| 0 | RATINT | REAL | | C031 | REFS | 10 | | | | | | |
| 0 | REASON | INTEGER | | COM7 | REFS | 8 | 16 | DEFINED | 44 | 61 | | |
| 2264 | RECOV | REAL | | COM1 | REFS | 6 | | | | | | |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 6 | 26 | | | | | |
| 76 | TC | REAL | | COM3 | REFS | 7 | 2*21 | | | | | |
| 2266 | TIME | REAL | | COM1 | REFS | 6 | 19 | 2*21 | 26 | 49 | 59 | 62 |
| | | | | | DEFINED | 59 | | | | | | |
| 202 | TIMI | REAL | | | REFS | 62 | DEFINED | 19 | | | | |
| 211 | U | REAL | | | REFS | 2*40 | DEFINED | 39 | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES | |
|---|---|---|---|---|
| FIFAU | | 3 | 27 | |
| GATHER | | 0 | 60 | |
| MISCYC | | 3 | 62 | |
| RANF | REAL | 1 | 24 | 39 |

| INLINE FUNCTIONS | TYPE | ARGS | DEF LINE | REFERENCES |
|---|---|---|---|---|
| AINT | REAL | 1 | INTRIN | 21 |

| STATEMENT LABELS | | DEF LINE | REFERENCES | | |
|---|---|---|---|---|---|
| 24 | 1 | 23 | 20 | | |
| 145 | 10 | 67 | 63 | | |
| 51 | 20 | 31 | 26 | | |
| 30 | 40 | 25 | 22 | 30 | 64 |
| 0 | 50 | 71 | 68 | | |
| 120 | 80 | 57 | 34 | 54 | |
| 123 | 81 | 58 | 36 | | |
| 105 | 82 | 48 | 37 | | |
| 103 | 83 | 46 | 40 | | |
| 117 | 100 | 55 | 49 | 50 | |

| LOOPS | LABEL | | INDEX | FROM-TO | LENGTH | PROPERTIES | EXT REFS | EXITS |
|---|---|---|---|---|---|---|---|---|
| 54 | 80 | * | J | 34 57 | 47B | | | |
| 152 | 50 | * | I | 68 71 | 10B | OPT | | EXITS |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS | NAME | (LENGTH) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 | IDIM | (1) | 1 | TABLE | (1200) | 1201 | PTR | (1) | |
| | | 1202 | CXTEN | (1) | 1203 | IDETEC | (1) | 1204 | RECOV | (1) | |
| | | 1205 | DELAY | (1) | 1206 | TIME | (1) | | | | |
| COM3 | 63 | 0 | ACFAU | (60) | 60 | ENDMIS | (1) | 61 | MEMSIZ | (1) | |
| | | 62 | TC | (1) | | | | | | | |
| COM7 | 1 | 0 | REASON | (1) | | | | | | | |
| CO10 | 2 | 0 | M | (1) | 1 | DURRA | (1) | | | | |
| CO31 | 2 | 0 | RATINT | (1) | 1 | ISYNC | (1) | | | | |
| CO60 | 1 | 0 | LSTFLT | (1) | | | | | | | |
| DETE | 3 | 0 | PDET | (1) | 1 | DETMAX | (1) | 2 | PDM | (1) | |
| FAILD2 | 1 | 0 | FOCO2 | (1) | | | | | | | |
| PHILM3 | 4 | 0 | NTR3 | (1) | 1 | NTR2 | (1) | 2 | NTR1 | (1) | |
| | | 3 | NTNR1 | (1) | | | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 215B | 141 |
| COMMON LENGTH | 2404B | 1284 |

```
      SUBROUTINE STATEC(NEXT)
C                     THIS VERSION:   MARCH 1976
C INTRODUCTION OF A  SPARE
C
C ******************************************************************
      COMMON/COM1/IDIM,TABLE(300,4),PTR,EXTEN,IDETEC,RECOV,DELAY,TIME
      COMMON/COM3/ACFAU(10,6),ENDMIS,MEMSIZ,TC
      COMMON/COM7/REASON
      COMMON/CO15/NOON(5),NWORK
      COMMON/CO18/IREP
      COMMON/CO28/MODSIM,NSPB
      COMMON/CO33/RMUP(5)
      COMMON/CO34/ISPARE(5)
      COMMON/CO35/CONDIT
      COMMON/CO36/RMISTH
      COMMON/CO37/ RMU
      COMMON/DETE/POET,DETMAX
      INTEGER PTR,REASON,EXTEN
C ******************************************************************
C
C DETERMINATION OF THE SPARE NUMBER
      DO 1 I=1,5
      IF (ISPARE(I).EQ.0) GO TO 1
      II=ISPARE(I)
      NOON(II)=1
      GO TO 2
    1 CONTINUE
C I IS THE SPARE NUMBER
    2 CONTINUE
      I=II
   10 CONTINUE
      IF (TABLE(PTR,1).GT.TIME+CONDIT) GO TO 20
      CALL FIFAU(IN,NEXT,1)
      IF(NEXT.EQ.5) RETURN
      GOTO 10
   20 CONTINUE
C A FAULT IN ANOTHER COMPUTER INTERRUPTS THE SWITCHING OF THE SPARE
      IDE=0
      DET=TIME+CONDIT
      DO 21 J=1,IDIM
      IF (ACFAU(J,3).EQ.0.) GO TO 22
      IUN=ACFAU(J,4)
      IF((IUN.EQ.I).OR.(NOON(IUN).EQ.0)) GOTO 21
      IF (ACFAU(J,6).GE.DET) GO TO 21
      DET=ACFAU(J,6)
      IDE=J
   21 CONTINUE
   22 CONTINUE
      IF (IDE.EQ.0) GO TO 23
      NOON(I)=0
      NWORK=NWORK-1
      TIME=DET
      NEXT=1
      IF (NWORK.EQ.2) NEXT=4
      RETURN
```

```
      23 CONTINUE
         MUL=0
         DO 40 J=1,IDIM
         IF (ACFAU(J,3).EQ.0.) GO TO 50
60       IUN=ACFAU(J,4)
         IF(NOON(IUN).EQ.0) GOTO 40
         IF(IUN.EQ.IREP) MUL=I
      40 CONTINUE
      50 CONTINUE
65    C DETERMINE IF THE SPARE IS GOOD
         IF(RANF(0.).LT.EXP(-RMU*(TIME-ENDMIS-RMISTM)/3600000.)) GO TO 51
      C THE SPARE IS BAD. ARE WE AWARE OF IT
         IF (RANF(0.) .GT.DETMAX)GO TO 52
      C DO NOT INJECT THIS SPARE
70       ISPARE(I)=0
         NSPB=NSPB-1
         NOON(I)=0
         TIME=TIME+CONDIT
         IF (NSPB.GT.0) GO TO 53
75       NWORK=NWORK-1
         NEXT=1
         IF (NWORK.EQ.2) NEXT=4
         IF (NWORK.EQ.1) NEXT=10
         RETURN
80    53 CONTINUE
         NEXT=12
         RETURN
      C THE SPARE IS BAD AND THIS IS NOT KNOWN. WE CREATE A NEW FAULT
      52 CONTINUE
85       PTR=PTR-1
         TABLE(PTR,1)=TIME+CONDIT
         TABLE(PTR,2)=10.*ENDMIS
         TABLE(PTR,3)=1.
         IF (RANF(0.).GT.RMUP(1)/RMU) TABLE(PTR,3)=3.
90       TABLE(PTR,4)=I
      51 CONTINUE
      C A FAULT IN IREP GIVES A FAULT IN THE SPARE
         IF (MUL.EQ.0) GO TO  54
         PTR=PTR-1
95       TABLE(PTR,1)=TIME+CONDIT
         TABLE(PTR,2)=TIME+CONDIT
         TABLE(PTR,3)=3.
         TABLE(PTR,4)=I
      54 CONTINUE
100      NEXT=1
         IF (NWORK.EQ.2) NEXT=8
         IF (NWORK.EQ.1) NEXT=10
         ISPARE(I)=0
         END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | STATEC | 1 | 34 | 55 | 79 | 82 | 104 | | |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACFAU | REAL | ARRAY | COM3 | REFS | 7 | 41 | 42 | 44 | 45 | 59 | 60 |
| 0 | CONDIT | REAL | | C035 | REFS | 14 | 32 | 39 | 73 | 86 | 95 | 96 |
| 2265 | DELAY | REAL | | COM1 | REFS | 6 | | | | | | |
| 253 | DET | REAL | | | REFS | 44 | 52 | DEFINED | 39 | 45 | | |
| 1 | DETMAX | REAL | | DETE | REFS | 17 | 68 | | | | | |
| 74 | ENDMIS | REAL | | COM3 | REFS | 7 | 66 | 87 | | | | |
| 2262 | EXTEN | INTEGER | | COM1 | REFS | 6 | 18 | | | | | |
| 247 | I | INTEGER | | | REFS | 23 | 24 | 43 | 50 | 62 | 70 | 72 |
| | | | | | | 90 | 98 | 103 | DEFINED | 22 | 30 | |
| 252 | IDE | INTEGER | | | REFS | 49 | DEFINED | 38 | 46 | | | |
| 2263 | IDETEC | INTEGER | | COM1 | REFS | 6 | | | | | | |
| 0 | IDIM | INTEGER | | COM1 | REFS | 6 | 40 | 58 | | | | |
| 250 | II | INTEGER | | | REFS | 25 | 30 | DEFINED | 24 | | | |
| 251 | IN | * INTEGER | | | REFS | 33 | | | | | | |
| 0 | IREP | INTEGER | | C018 | REFS | 10 | 62 | | | | | |
| 0 | ISPARE | INTEGER | ARRAY | C034 | REFS | 13 | 23 | 24 | DEFINED | 70 | 103 | |
| 255 | IUN | INTEGER | | | REFS | 2*43 | 61 | 62 | DEFINED | 42 | 60 | |
| 254 | J | INTEGER | | | REFS | 41 | 42 | 44 | 45 | 46 | 59 | 60 |
| | | | | | DEFINED | 40 | 58 | | | | | |
| 75 | MEMSIZ | INTEGER | | COM3 | REFS | 7 | | | | | | |
| 0 | MODSIM | INTEGER | | C028 | REFS | 11 | | | | | | |
| 256 | MUL | INTEGER | | | REFS | 93 | DEFINED | 57 | 62 | | | |
| 0 | NEXT | INTEGER | | F.P. | REFS | 33 | 34 | DEFINED | 1 | 53 | 54 | 76 |
| | | | | | | 77 | 78 | 81 | 100 | 101 | 102 | |
| 0 | NOON | INTEGER | ARRAY | C015 | REFS | 9 | 43 | 61 | DEFINED | 25 | 50 | 72 |
| 1 | NSPB | INTEGER | | C028 | REFS | 11 | 71 | 74 | DEFINED | 71 | | |
| 5 | NWORK | INTEGER | | C015 | REFS | 9 | 51 | 54 | 75 | 77 | 78 | 101 |
| | | | | | | 102 | DEFINED | 51 | 75 | | | |
| 0 | PDET | REAL | | DETE | REFS | 17 | | | | | | |
| 2261 | PTR | INTEGER | | COM1 | REFS | 6 | 18 | 32 | 85 | 86 | 87 | 88 |
| | | | | | | 89 | 90 | 94 | 95 | 96 | 97 | 98 |
| | | | | | DEFINED | 85 | 94 | | | | | |
| 0 | REASON | INTEGER | | COM7 | REFS | 8 | 18 | | | | | |
| 2264 | RECOV | REAL | | COM1 | REFS | 6 | | | | | | |
| 0 | RMISTM | REAL | | C036 | REFS | 15 | 66 | | | | | |
| 0 | RMU | REAL | | C037 | REFS | 16 | 66 | 89 | | | | |
| 0 | RMUP | REAL | ARRAY | C033 | REFS | 12 | 89 | | | | | |
| 1 | TABLE | REAL | ARRAY | COM1 | REFS | 6 | 32 | DEFINED | 86 | 87 | 88 | 89 |
| | | | | | | 90 | 95 | 96 | 97 | 98 | | |
| 76 | TC | REAL | | COM3 | REFS | 7 | | | | | | |
| 2266 | TIME | REAL | | COM1 | REFS | 6 | 32 | 39 | 66 | 73 | 86 | 95 |
| | | | | | | 96 | DEFINED | 52 | 73 | | | |

| EXTERNALS | TYPE | ARGS | | REFERENCES | | |
|---|---|---|---|---|---|---|
| EXP | REAL | 1 | LIBRARY | 66 | | |
| FIFAU | | 3 | | 33 | | |
| RANF | REAL | 1 | | 66 | 68 | 89 |

| STATEMENT LABELS | | DEF LINE | REFERENCES | | |
|---|---|---|---|---|---|
| 15 | 1 | 27 | 22 | 23 | |
| 17 | 2 | 29 | 26 | | |
| 21 | 10 | 31 | 35 | | |
| 35 | 20 | 36 | 32 | | |
| 61 | 21 | 47 | 40 | 43 | 44 |
| 63 | 22 | 48 | 41 | | |
| 100 | 23 | 56 | 49 | | |
| 120 | 40 | 63 | 58 | 61 | |
| 122 | 50 | 64 | 59 | | |
| 204 | 51 | 91 | 66 | | |
| 164 | 52 | 84 | 68 | | |
| 161 | 53 | 80 | 74 | | |
| 216 | 54 | 99 | 93 | | |

| LOOPS | LABEL | | INDEX | FROM-TO | LENGTH | PROPERTIES | |
|---|---|---|---|---|---|---|---|
| 10 | 1 | * | I | 22 27 | 78 | INSTACK | EXITS |
| 45 | 21 | * | J | 40 47 | 168 | OPT | EXITS |
| 106 | 40 | * | J | 58 63 | 148 | OPT | EXITS |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LENGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| COM1 | 1207 | 0 IOIH (1) | 1 TABLE (1200) | 1201 PTR (1) | | | |
| | | 1202 EXTEN (1) | 1203 IDETEC (1) | 1204 RECOV (1) | | | |
| | | 1205 DELAY (1) | 1206 TIME (1) | | | | |
| COM3 | 63 | 0 ACFAU (60) | 60 ENDMIS (1) | 61 MEMSIZ (1) | | | |
| | | 62 TC (1) | | | | | |
| COM7 | 1 | 0 REASON (1) | | | | | |
| CO15 | 6 | 0 NOON (5) | 5 NWORK (1) | | | | |
| CO18 | 1 | 0 IREP (1) | | | | | |
| CO28 | 2 | 0 MODSIM (1) | 1 NSPB (1) | | | | |
| CO33 | 5 | 0 RMUP (5) | | | | | |
| CO34 | 5 | 0 ISPARE (5) | | | | | |
| CO35 | 1 | 0 CONDIT (1) | | | | | |
| CO36 | 1 | 0 RMISTH (1) | | | | | |
| CO37 | 1 | 0 RMU (1) | | | | | |
| DETE | 2 | 0 PDET (1) | 1 DETMAX (1) | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 2578 | 175 |
| COMMON LENGTH | 24178 | 1295 |

```
      SUBROUTINE TFISO(LOC,IP1)
      COMMON/TFRFCB/TF0(6),TF1(8),TF2(10),TF2X(10,4),TF3(2),TF3X(2,4)
     1,TF4(4),TF4X(4,6),TF5(4),TF5X(4,4),TF5Y(4,2),TF6(4),TF6X(4,3)
      DIMENSION IP(5)
      IGRP=ISTEPD(6,TF0)
      IP(2)=IGRP
      IP(1)=IP1
      IP(5)=0
      GOTO(1000,2000,3000,4000,5000,6000),IGRP
 1000   CONTINUE
          IP(3)=ISTEPD(8,TF1)
          IP(4)=0
          GOTO 9000
 2000   CONTINUE
          MDM=ISTEPD(10,TF2)
          IP(3)=MDM
          IP(4)=MSTEPD(MDM,TF2X,10,4)-1
          GOTO 9000
 3000   CONTINUE
          MDM=ISTEPD(2,TF3)
        IP(3)=MDM
          IP(4)=MSTEPD(MDM,TF3X,2,4)
          GOTO 9000
 4000   CONTINUE
          MDM=ISTEPD(4,TF4)
          IP(3)=MDM
          IP(4)=MSTEPD(MDM,TF4X,4,6)
          GOTO 9000
 5000   CONTINUE
          IDEV=ISTEPD(4,TF5)
          IP(5)=IDEV
          IP(3)=MSTEPD(IDEV,TF5X,4,4)-1
        IP(4)=MSTEPD(IDEV,TF5Y,4,2)
          GOTO 9000
 6000   CONTINUE
          MDM=ISTEPD(4,TF6)
          IP(3)=MDM
          IP(4)=MSTEPD(MDM,TF6X,4,3)
          GOTO 9000
 9000 CONTINUE
      CALL PACK(IP,LOC)
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | TFISO | 1 | 42 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 175 | IDEV | INTEGER | | | REFS | 31 | 32 | 33 | DEFINED | 30 | | |
| 173 | IGRP | INTEGER | | | REFS | 6 | 9 | DEFINED | 5 | | | |
| 176 | IP | INTEGER | ARRAY | | REFS | 4 | 41 | DEFINED | 6 | 7 | 8 | 11 |
| | | | | | 12 | 16 | 17 | 21 | 22 | 26 | 27 | 31 |
| | | | | | 32 | 33 | 37 | 38 | | | | |
| 0 | IP1 | INTEGER | | F.P. | REFS | 7 | DEFINED | 1 | | | | |
| 0 | LOC | INTEGER | | F.P. | REFS | 41 | DEFINED | 1 | | | | |
| 174 | MDM | INTEGER | | | REFS | 16 | 17 | 21 | 22 | 26 | 27 | 37 |
| | | | | | 38 | DEFINED | 15 | 20 | 25 | 36 | | |
| 0 | TF0 | REAL | ARRAY | TFRFCB | REFS | 2 | 5 | | | | | |
| 6 | TF1 | REAL | ARRAY | TFRFCB | REFS | 2 | 11 | | | | | |
| 16 | TF2 | REAL | ARRAY | TFRFCB | REFS | 2 | 15 | | | | | |
| 30 | TF2X | REAL | ARRAY | TFRFCB | REFS | 2 | 17 | | | | | |
| 100 | TF3 | REAL | ARRAY | TFRFCB | REFS | 2 | 20 | | | | | |
| 102 | TF3X | REAL | ARRAY | TFRFCB | REFS | 2 | 22 | | | | | |
| 112 | TF4 | REAL | ARRAY | TFRFCB | REFS | 2 | 25 | | | | | |
| 116 | TF4X | REAL | ARRAY | TFRFCB | REFS | 2 | 27 | | | | | |
| 146 | TF5 | REAL | ARRAY | TFRFCB | REFS | 2 | 30 | | | | | |
| 152 | TF5X | REAL | ARRAY | TFRFCB | REFS | 2 | 32 | | | | | |
| 172 | TF5Y | REAL | ARRAY | TFRFCB | REFS | 2 | 33 | | | | | |
| 202 | TF6 | REAL | ARRAY | TFRFCB | REFS | 2 | 36 | | | | | |
| 206 | TF6X | REAL | ARRAY | TFRFCB | REFS | 2 | 38 | | | | | |

| EXTERNALS | TYPE | ARGS | REFERENCES | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ISTEPD | INTEGER | 2 | 5 | 11 | 15 | 20 | 25 | 30 | 36 |
| MSTEPD | INTEGER | 4 | 17 | 22 | 27 | 32 | 33 | 38 | |
| PACK | | 2 | 41 | | | | | | |

| STATEMENT LABELS | | DEF LINE | REFERENCES | | | | | |
|---|---|---|---|---|---|---|---|---|
| 24 | 1000 | 10 | 9 | | | | | |
| 30 | 2000 | 14 | 9 | | | | | |
| 37 | 3000 | 19 | 9 | | | | | |
| 45 | 4000 | 24 | 9 | | | | | |
| 53 | 5000 | 29 | 9 | | | | | |
| 64 | 6000 | 35 | 9 | | | | | |
| 72 | 9000 | 40 | 13 | 18 | 23 | 28 | 34 | 39 |

| COMMON BLOCKS | LENGTH | MEMBERS - BIAS NAME(LFNGTH) | | | | | |
|---|---|---|---|---|---|---|---|
| TFRFCB | 146 | 0 TF0 (6) | 6 TF1 (8) | 14 TF2 (10) | | | |
| | | 24 TF2X (40) | 64 TF3 (2) | 66 TF3X (8) | | | |
| | | 74 TF4 (4) | 78 TF4X (24) | 102 TF5 (4) | | | |
| | | 106 TF5X (16) | 122 TF5Y (8) | 130 TF6 (4) | | | |
| | | 134 TF6X (12) | | | | | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 2038 | 131 |
| COMMON LENGTH | 222B | 146 |

```
      SUBROUTINE TSTRNF(IPLACE)
      DIMENSION MP1(2,20)
      DIMENSION MP2(6,20)
      DIMENSION IPLACE(5)
      DATA IP1P/0/,IP2P/0/,IP1C/0/,IP2C/0/
      DATA MP1/40*0/,MP2/120*0/
      IP1=IPLACE(1)
      IP2=IPLACE(2)
      IF(IP1.LE.0) GOTO 10
         IF(IP1.EQ.IP1P) IP1C=IP1C+1
         IF(IP1.NE.IP1P) IP1C=1
         IF(IP2.EQ.IP2P) IP2C=IP2C+1
         IF(IP2.NE.IP2P)IP2C=1
         IP1P=IP1
         IP2P=IP2
         MP1(IP1,IP1C)=MP1(IP1,IP1C)+1
         MP2(IP2,IP2C)=MP2(IP2,IP2C)+1
         RETURN
   10 CONTINUE
      PRINT 1000,((MP1(I,J),J=1,20),I=1,2)
      PRINT 1000,((MP2(I,J),J=1,20),I=1,6)
 1000 FORMAT(/1X,20I6)
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|
| 2 | TSTRNF | 1 | 18 | 23 |

| VARIABLES | SN | TYPE | RELOCATION | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 120 | I | INTEGER | | | REFS | 20 | 21 | DEFINED | 20 | 21 | | |
| 0 | IPLACE | INTEGER | ARRAY | F.P. | REFS | 4 | 7 | 8 | DEFINED | 1 | | |
| 116 | IP1 | INTEGER | | | REFS | 9 | 10 | 11 | 14 | 2*16 | | |
| | | | | | DEFINED | 7 | | | | | | |
| 112 | IP1C | INTEGER | | | REFS | 10 | 2*16 | DEFINED | 5 | 10 | 11 | |
| 110 | IP1P | INTEGER | | | REFS | 10 | 11 | DEFINED | 5 | 14 | | |
| 117 | IP2 | INTEGER | | | REFS | 12 | 13 | 15 | 2*17 | DEFINED | 8 | |
| 113 | IP2C | INTEGER | | | REFS | 12 | 2*17 | DEFINED | 5 | 12 | 13 | |
| 111 | IP2P | INTEGER | | | REFS | 12 | 13 | DEFINED | 5 | 15 | | |
| 121 | J | INTEGER | | | REFS | 20 | 21 | DEFINED | 20 | 21 | | |
| 122 | MP1 | INTEGER | ARRAY | | REFS | 2 | 16 | 20 | DEFINED | 6 | 16 | |
| 172 | MP2 | INTEGER | ARRAY | | REFS | 3 | 17 | 21 | DEFINED | 6 | 17 | |

| FILE NAMES | MODE | | | |
|---|---|---|---|---|
| OUTPUT | FMT | WRITES | 20 | 21 |

| STATEMENT LABELS | | DEF LINE | REFERENCES | |
|---|---|---|---|---|
| 52 | 10 | 19 | 9 | |
| 114 | 1000 FMT | 22 | 20 | 21 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES | |
|---|---|---|---|---|---|---|
| 55 | | * I | 20 | 118 | EXT REFS | NOT INNER |
| 56 | | * J | 20 | 68 | EXT REFS | |
| 72 | | * I | 21 | 128 | EXT REFS | NOT INNER |
| 73 | | * J | 21 | 78 | EXT REFS | |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 3668 | 246 |

```
      SUBROUTINE UNIF(FAU,N,RMINT,RMAXT)
C                                        THIS VERSION: 25 FEBRUARY 1974
C IT GENERATES N UNIFORMLY DISTRIBUTED DURATIONS
C FAU     ARRAY OF DURATIONS
C N       DIMENSION OF FAU
C RMINT   MINIMUM DURATION
C RMAXT   MAXIMUM DURATION
C
C********** ********* ********* ********** ********* ********** ********
      DIMENSION FAU(N)
C ***************************************************************
C
      DO 20 K=1,N
      Y=RANF(0.)
      FAU(K)=(RMAXT-RMINT)*Y+RMINT
   20 CONTINUE
      END
```

SYMBOLIC REFERENCE MAP

| NTRY POINTS | DEF LINE | REFERENCES |
|---|---|---|
| 2 UNIF | 1 | 17 |

| ARIABLES | SN | TYPE | | RELOCATION | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 FAU | | REAL | ARRAY | F.P. | REFS | 10 | DEFINED | 1 | 15 |
| 32 K | | INTEGER | | | REFS | 15 | DEFINED | 13 | |
| 0 N | | INTEGER | | F.P. | REFS | 10 | 13 | DEFINED | 1 |
| 0 RMAXT | | REAL | | F.P. | REFS | 15 | DEFINED | 1 | |
| 0 RMINT | | REAL | | F.P. | REFS | 2*15 | DEFINED | 1 | |
| 33 Y | | REAL | | | REFS | 15 | DEFINED | 14 | |

| XTERNALS | TYPE | ARGS | REFERENCES |
|---|---|---|---|
| RANF | REAL | 1 | 14 |

| TATEMENT LABELS | DEF LINE | REFERENCES |
|---|---|---|
| 0 20 | 16 | 13 |

| OOPS | LABEL | INDEX | FROM-TO | LENGTH | PPOPERTIES |
|---|---|---|---|---|---|
| 15 | 20 | * K | 13 16 | 10B | EXT REFS |

TATISTICS
PROGRAM LENGTH 440 36

```
      SUBROUTINE UNPACK(LOC,PLACE)
      INTEGER PLACE
      DIMENSION PLACE(5)
      DO 10 J=1,5
5       PLACE(J)=AND(63,LOC)
        LOC=SHIFT(LOC,-6)
   10 CONTINUE
      RETURN
      END
```

SYMBOLIC REFERENCE MAP

| ENTRY POINTS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 2 | UNPACK | 1 | 8 |

| VARIABLES | SN | TYPE | | RELOCATION | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 25 | J | INTEGER | | | REFS | 5 | DEFINED | 4 | | |
| 0 | LOC | INTEGER | | F.P. | REFS | 5 | 6 | DEFINED | 1 | 6 |
| 0 | PLACE | INTEGER | ARRAY | F.P. | REFS | 2 | 3 | DEFINED | 1 | 5 |

| INLINE FUNCTIONS | TYPE | ARGS | | DEF LINE | REFERENCES |
|---|---|---|---|---|---|
| AND | NO TYPE | 2 | INTRIN | | 5 |
| SHIFT | NO TYPE | 2 | INTRIN | | 6 |

| STATEMENT LABELS | | DEF LINE | REFERENCES |
|---|---|---|---|
| 0 | 10 | 7 | 4 |

| LOOPS | LABEL | INDEX | FROM-TO | LENGTH | PROPERTIES |
|---|---|---|---|---|---|
| 17 | 10 | J | 4 7 | 38 | INSTACK |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 338 | 27 |